commodore 64

Dr. Bakó András

Alkalmazói softwarek

1

Dr. Bakó András

Alkalmazói softwarek

Felelős kiadó; Novotrade RT. igazgatója Rényi Gábor

AGROÉPSZER GT. Nyomda Tsz: 86/006

Bevezetés

A könyv megirásának célja a COMMODORE 64 gépekre meglevő gyári softver anyagok kezelésének könnyebbé tétele a magyar felhasználó számára. A közölt anyag a gyári kézikönyvek és az utasitások kipróbálása során nyert tapasztala - tok alapján készült. A megirás során gyári kézikönyvekben levő nem tematikus tárgyalást igyekeztem tematikus rendbe szedni. Igy egy paragrafusba kerültek az összetartozó utasitások is.

Az egyes softver anyagok leirásánál ezért alakultak ki a téma szerint összetartozó fejezetek:

Programozást segitő eljárások
Programbelövést segitő rutinok
Grafikus eljárások
Input-output utasitások stb.

Az alábbi anyagok leirását adom meg:

a.) HELP PLUS

A gyári softver anyag hatékony segédeszköz a programok irásához, belövéséhez. A SIMON'S BASIC azonban az itt levő utasitások nagy részét tartalmazza, és több mint kétszerannyi további utasitást, parancsot is. A rendszer parancsait kiptóbáltam, és mindegyik a leirásnak megfelelően müködött. Nem részleteztem az ASSEMBLER forditó használatát, mivel az a felhasználók csak kis részét érinti. Az ezen rész használatához javasolom az eredeti leirás megfelelő részének áttanulmányozását (utolsó néhány oldal).

b.) SIMON'S BASIC

Az egyik leghatékonyabb programozást segitő softver anyag. A gyári leirások (angol és német) egymástól több helyen eltérnek. Némelyik nem ugy müködik,

ahogy az eredeti leirás és a mintapélda mutatja. A leirást az egyes utasítások, parancsok kipróbálása után végeztem el. A jobb megértés illusztrálására igyekeztem mindenütt példa programot is késziteni. Remélem ezek segitik az anyagok hatékony felhasználását is.

c.) SUPER GRAPHIK 64

A rendszer két változata létezik: a szalagos és a lemezes változat. A szalagos változat utasitásait részletesen leirom, és mintapéldákat is közlök. A lemezes változatból eddig csak néhány érkezett hazánkba és az utasitások nagy része hasonló a szalagos változathoz. Ezért a lemezes változatból csak a többlet utasitásokat irom le részletesen, és adok rá mintapéldát. Ez a többlet: a zenei utasitások és a 80x50-es grafika.

d.) C64 AIDS

A rendszer parancsai a programirást és programbelövést segitik. Tulajdonképpen a legkevesebbet tudják a 3 hasonló jellegü softver termékből (a, b, c)

A könyv végén összefoglalom a SIMON'S BASIC hibaüzeneteit, és itt közlöm az egyes utasitások, parancsok összefoglalását is.

Mint minden ilyen jellegü kézikönyvnél, itt is előfordulhatnak pontatlanságok, elirások. Az ilyenekért az olvasó szives elnézését kérem. Remélem, a gép hatékonyabb kihasználását szerény irásommal elő tudtam segiteni. A további felhasználói rendszerek közlését a következő kötet fogja tartalmazni (szövegszerkesztés, adatbank softver stb.).

HELP PLUS

Bevezetés

A HELP PLUS egy a C64 számitógép felhasználását elősegitő utasitásgyüjtemény. Az alábbi funkcionálisan elkülönithető részekből áll:

- programirást, listázást segitő parancsok,
- programbelövést segitő parancsok,
- egyéb rendszerparancsok,
- disassembler parancsok,
- disk kezelés parancsai.

A továbbiakban végigvesszük a fenti részekhez tartozó parancsokat.

1. Programozást, listázást segítő parancsok

1.1. Automatikus sorszámkiirás

A programozást segiti elő a GETLINE parancs. Ezzel a program első utasitásának sorszámát (utasitáscimkéjét) és az utasitássorszámok közötti növekményt adhatjuk meg.

Alakja:

#G a, b

ahol	#G	a GETLINE parancs röviditett alakja
	a	az első utasitás kezdősorszáma
	b	az utasitás sorszámok közötti különbség (növekmény)

Hatása: a gép egy-egy utasitás begépelése után automatikusan kiirja a következő utasitás sorszámát.

Megjegyzések:

1. A következő sorszám a RETURN gomb megnyomása után iródik ki.

2. A parancs hatása befejeződik, ha a következő sorszámot a DEL és RETURN gombbal töröljük vagy a SHIFT és RETURN gombot egyszerre megnyomjuk.

3. Az a és b paraméterekre előirt korlátozások:

   b 1 és 255 közötti értéket vehet fel,

   a 63999-nél kisebb egész szám lehet.

4. Ha a és b nincs megadva, akkor

   a = 100, b = 10

5. Ha az automatikus sorszámozás eléri a 64000 értéket, akkor tovább folyik, de az aktuális sorszámból levonja a fenti értéket.

1.2. Szabad memóriahely kérése

A programozás vagy futtatás során szükséges a még szabad memóriaterület ismerete. Ezt adja meg a * parancs.
Alakja:

*

Hatása: megegyezik a PRINT FRE (U) utasítással.

1.3. Program formázott listázása

A gépben levő program megadott részenként, oldalanként történő listázására is lehetőség van a SEITENLIST paranccsal.
Alakja:

#L n

ahol #L a parancs röviditett alakja
n egy egész szám, a kezdő utasitássorszám.

Hatása: az n. sorszámu utasitástól kezdve kiir a képernyőre a gépben levő programból annyit, amennyi a képernyőre fér.

Megjegyzések:

1. Az n paraméter elmaradhat. Ekkor a gép a program első utasitásától kezdve listáz.

2. A RETURN billentyü hatására a következő képernyőnyi programrészt irja ki.

3. Ha visszafelé, csökkenő számsorrendben akarunk listázni, akkor a billentyüt kell megnyomnunk.

4. A parancs hatását a RUN/STOP billentyűvel szüntethetjük meg.

1.4. Program átsorszámozása

Egy gépben levő program utasitás sorszámainak megváltoztatását, ujra sorszámozását végzi a RENUMBER parancs.
Alakja:

#R a, b

ahol #R az utasitás parancsszava,
a kezdő sorszám (a ≤ 1 ≤ 63999),
b sorszámok közötti növekmény (1 ≤ b ≤ 255).

Hatása: a gépben levő programot átsorszámozza a -tól kezdve b -esével.

Megjegyzések:

1. A paraméterek el is maradhatnak. Ilyenkor

   a = 100, b= 10

   értékkel dolgozik a gép.

2. Az utasitás sorszámot tartalmazó utasitásokban a sorszámok automatikusan átsorszámozódnak.
   Ilyen utasitások a következők:

   GO TO, GOSUB, ON...,
   THEN, RUN

3. Az átszámozásnál tul nagy a illetve b esetén illegális sorszámok is adódhatnak, amelyek 63999-nél nagyobbak. Ezért célszerű a #U-val leellenőrizni a programot.

1.5. Program sűritése

A felesleges üres helyek és sorszámozások összevonása a programfutást gyorsitja. Erre szolgál COMPAKTOR parancs.

Alakja:

#C a

ahol #C a parancs röviditett alakja,

a egész szám, az egy sorban elhelyezhető karakterek száma ($1 \leq a \leq 240$).

Hatása: a parancs a gépben levő program utasitásait összesűriti ugy, hogy egy sorba lehetőleg a megadott a számnyi karakter legyen.

Megjegyzések:

1. A felesleges üres helyeket és megjegyzéseket a gép automatikusan kihagyja (REM-ek).

2. Azon utasitásokat, ahova ugrás történik külön sorba teszi.

3. Ha van UNDEF'D STATEMENT, akkor nem hajtódik végre a sűrités.

4. Az átszámozás - a fentieket figyelembe véve - ugy történik, mintha a

   #R 1,2

   parancsot használtuk volna.

5. A sűritett program futása gyorsabb, de a program továbbfejlesztése nagyon problematikus a nehéz olvashatóság miatt.

1.6. Programok összemásolása

Részekben megirt programok külső tárolóról való összemásolását végzi az APPEND parancs.

Alakja:

#A "n" , d

ahol #A a parancs kulcsszava,
n a másolandó program neve,
d egységszám, ahol a másolandó program van.

Hatása: a gépben levő program végétől kezdve bemásolja a megadott d egységszámu háttértárolóról az n nevü programot.

Megjegyzések:

1. Ha d hiányzik, akkor a gép a kazettás egységet veszi háttértárolóként.

2. A másolás előtt gondoskodni kell a programrész különböző számozásáról.

3. A gépben levő gépi kódu program a másolás során felülíródik.

1.7. Egy programrészlet megkeresése

A programirás vagy programbelövés során sokszor szükséges egy karaktersorozat (konstans, változó, kulcsszó, stb.) megkeresése. Ezt végzi a FIND parancs.

Alakja:

#F s

ahol F a parancs kulcsszava,
s a programban előforduló tetszésszerinti karaktersorozat.

Hatása: a gépben levő programban megkeresi és képernyőre kilistázza az összes utasitást, az alábbi szabályok szerint.

Szabályok:

1. Utasitás megkeresése esetén a teljes alapszót ki kell irni.
2. Függvény esetén a függvény alapszava után legalább a nyitózárójelet ki kell irni.
3. String egy részletet a # F" ... paranccsal lehet megkeresni.
4. A fenti három pontban emlitettek kivételével tetszés szerinti programrészletet meg lehet adni.
5. Fontos megjegyeznünk, hogy a kereséshez az üres helynek is fontos szerepe van.
6. Az utasitások listázását a SHIFT gomb lenyomásával megállithatjuk és továbbithatjuk.
7. A STOP gomb lenyomása megszakitja a parancs végrehajtását.
8. A CMD megfelelő beállitásával a listázás történhet például nyomtatóra is a LIST parancshoz hasonlóan.

1.8. Programsorok törlése

Egy vagy több programsor törlését végzi a DELETE parancs.

Alakja:

a.) #D n

b.) #D -n

c.) #D n-

d.) #D $n_1 - n_2$

ahol #D a törlés parancs,

n, n_1, n_2 utasitássorszámok.

Hatása: a gépben levő program

a.) n. sorszámu sorát eltörli,

b.) sorait eltörli egészen az n. sorszámu utasitással bezárólag,

c.) n. sorától kezdve eltörli az utasitásokat,

d.) n_1. sorától kezdve az n_2. sorig kitörli az utasitásokat.

Megjegyzés:

A parancs után a CONT parancsot nem használhatjuk.

2. Programbelövést segitő parancsok

2.1. Programhiba megadása megszakitás esetén

A program futásának megszakitása esetén a hiba beazonositásának munkáját segiti a HELP parancs.

Alakja:

#H

Hatása: a parancs kiadása esetén a program futása után a következő történik:

a.) ha normálisan fejeződött be a program, akkor kiirja az utoljára végrehajtott utasitás-sort,

b.) ha a futás megszakitása programhiba miatt történt, akkor a hibás sor kiiródik, a hibát okozó részt "reverse"-ben adja meg a gép.

Megjegyzések:

A HELP parancsot közvetlenül a program befejezése után kell kiadni. Egyébként a parancs hatástalan.

2.2. Program futásának nyomonkövetése

Az utasitások végrehajtási sorrendjét adja meg a TRACE és a SINGLE STEP utasitás.

2.2.1. TRACE utasitás

Alakja:

#T

Hatása: a program utasitásainak sorszámait végrehajtási sorrendjük szerint kiirja a képernyő legfelső sorába.

Megjegyzések:

1. A RUN, GOSUB, CONT utasitások hatására elinditott program utasitásai kiiródnak a képernyő 2. sorába.
2. Az utolsó 6 utasitás sorszáma a képernyő legelső sorában jelenik meg.
3. A program végrehajtása lassu, hogy a kiirt adatokat értékelni tudjuk. A SHIFT gomb lenyomásakor a program végrehajtása gyorsul.
4. Az eredmények kiirásánál figyelembe kell venni a TRACE müködését, amennyiben a parancsot használni akarjuk.
5. Vigyázzunk az egyes INPUT utasitásokra, mivel a TRACE ezt is módosithatja. Ezt a kurzor üres sorba vivésével küszöbölhetjük ki. A parancs hatását a következő parancs begépelésével szüntethetjük meg:

#E

2.2.2. Programvégrehajtás soronként

A program utasitásonkénti végrehajtását végzi a SINGLE STEP parancs.

Alakja:

#S

Hatása: a program utasításait egymásután végrehajtja ugy, hogy minden lépés után felfüggeszti a futást. A továbbinditás a SHIFT billentyüvel történik.

Megjegyzések:

1. A kiirás formája megegyezik a TRACE utasitásnál elmondottakkal.
2. A program folyamatosan futtatható a SHIFT LOCK billentyüvel.
3. A parancs hatása megszüntethető a

#E

paranccsal

2.3. Illegális hivatkozási helyek

A nem megengedett hivatkozási helyek felülvizsgálatát végzi a #U utasitás.

Alakja:

#U

Hatása: megvizsgálja, hogy a gépben levő programban hol vannak olyan hivatkozások (GO TO, PERFORM stb), amelyekhez tartozó utasitások sorszámait nem tartalmazza a program. Az ilyen helyek sorszámát is kilistázza képernyőre.

2.4. Programváltozók kiirása

A HELP + segitségével egy program egyszerü és indexes változóinak értékeire is rákérdezhetünk.

2.4.1. Egyszerü változók értékeinek kiirása

Egyszerü változók értékének a kiirására szolgál a VARIABLE DUMP parancs.

Alakja:

#V

Hatása: az éppen gépben levő program egyszerü (nem indexes) változóinak tartalmát irja ki a parancs hatására.

Megjegyzések:

1. A kiirás ugy történik, hogy a képernyőn egymásután változóként külön sorba kiirja a változók értékét

$$V_i = \text{érték}_i$$

formában.

2. A kiirás lassitása szükséges, amennyiben nem férnek el egy képernyőre. Ehhez nyomjuk meg a CTRL gombot.

2.4.2. Tömbök tartalmának kiirása

Egy program tömbjei elemeinek értékét irja ki a MATRIZEN DUMP parancs.

Alakja:

#M

Hatása: Az éppen gépben levő program indexes változóinak értékét irja ki elemenként az egyszerü változókhoz hasonlóan.

2.5. Konstansok konvertálása

A memóriatartalom elemzéséhez, bizonyos memóriacimek meghatározásához sokszor szükség van a tizes és tizenhatos számrendszerbeli számok kölcsönös konvertálásához.

2.5.1. Konvertálás tizenhatosból tizesbe

Egy tizenhatos számrendszerbeli számot konvertál tizes számrendszerbeli számra a !# utasitás.

Alakja:

!# k1k2k3k4

ahol !# a konvertáló utasitás

k1,k2,k3,k4 tizenhatos számrendszerbeli számjegyek.

Hatása: a parancs kiadása és a RETURN gomb lenyomása után a gép a képernyőre kiirja a szám 10-es számrendszerbeli értékét.

Megjegyzések:

1. A tizenhatos számrendszerbeli szám legfeljebb 4 jegyü lehet (tehát 1, 2, és 3 jegy is szerepelhet).
2. Ha 4 jegynél többet adunk meg, az

?ILLEGAL QUANTITY ERROR IN...

hibaüzenetet kapunk.

2.5.2. Konvertálás tizesből tizenhatosba

Tizes számrendszerbeli számot tizenhatosba a !$ utasitással alakithatunk át.

Alakja:

!$ $n_1n_2n_3n_4n_5$

ahol !$ a konvertálási utasitás,

$n_1n_2n_3n_4n_5$ tizes számrendszerbeli számjegyek.

Megjegyzések:

1. A számjegyek száma legfeljebb 5 lehet. A legnagyobb konvertálható érték 65535 lehet tizesben.
2. Nagyobb érték megadása esetén a fenti hibaüzenet iródik ki.

3. Egyéb parancsok

3.1. Törölt program aktivizálása

A NEW paranccsal a memóriából kitörölt program visszaállitását végzi a RENEW parancs.

Alakja:

#B

Hatása: a fent elmondott.

Megjegyzések:

1. A parancs a BASIC programot megszabaditja a hozzákapcsolt gépi programtól.
2. Ha a gép valamilyen okból "lemerevedik" (azaz a kurzor eltűnik), akkor a gép aktivizálása a STOP/RESTORE gombok egyszerre történő megnyomásával lehetséges. Az esetleg elvesztett programot ilyenkor ezzel a paranccsal kaphatjuk vissza.

3.2. CMD utasitás

Az OPEN és a CMD utasitás együttes hatását eredményezi a (parancs.

Alakja:

(d

ahol (az utasitás "kulcsszava",

d egységszám (1 és 255 közötti érték).

Megjegyzések:

1. Ha a d elmarad, akkor d = 4 értéket vesz fel a gép automatikusan,

Ekkor a parancs hatása megegyezik a következő parancsok végrehajtásával:

OPEN 4,4
CMD 4

2. Ha a d értéke más, akkor az 1. pontban leírt parancsokkal egyezik meg, csak az egységszám lesz értelemszerűen más.

Az igy megnyitott file lezárására szolgál a) parancs.
Alakja:

tehát a parancsnak nincs paramétere.

3.3. Egységszám megváltoztatása

Egységszám megváltoztatására szolgál a £ parancs.
Alakja:

£ d

ahol £ a parancs
d egységszám (4-31 közötti érték).

Hatása: HELP PLUS DOS és ASSEMBLER parancsok részére megváltoztatja a 8-as egységszámot d-re.

3.4. HELP PLUS befejezése

A HELP + "kikapcsolására" szolgál a KILL parancs.
Alakja:

#K

Hatása: a HELP + kikapcsolódik, a gépben levő programot nem rontja el.

4. Disassembler parancsok

Az itt felsorolt parancsok a BASIC interpretert kikapcsolják és a gép monitor állapotban dolgozik.

4.1. Disassembler aktivizálása

A BASIC értelmező kikapcsolását és a disassembler "bekapcsolását" végzi a] parancs.

Alakja:

]

A képernyőn a következő jelenik meg alapállapotban:

0000 2F

Megjegyzés:

1. Az első négy jegy hexadecimális és a két byte-os tárcimet adja meg.
2. Az utolsó két jegy hexadecimális és a tárcimhez tartozó 1 byte hosszuságu memóriatartalmat jelenti.
3. A hexadecimális (tizenhatos számrendszerbeli) számjegyek: 0 - 9 megegyezik a tizessel. További számjegyek:

   A:10, B:11, C:12, D:13
   E:14, F:15

4. A disassembler 3 állapotban lehet, és minden állapotban további parancsokat ismer.

4.2. Cim és tartalom változtatása

Ebben az állapotban a cimeket illetve tartalmakat változtathatjuk meg. Bekapcsoláskor ebbe az állapotba kerül a gép.

4.2.1. Cimváltoztatás

Cimváltoztató parancs a +.
Alakja:

+

A billentyü lenyomása után a 0 - F "számjegy" billentyük valamelyikét nyomhatjuk meg. A lenyomott billentyü a négyjegyü cim utolsó jegyének helyére kerül az utolsó 3 jegy előrecsusztatása után. Az első számjegy ekkor elvész.

A további számjegy billentyük lenyomásával a fentiekben elmondottak történnek.

4.2.2. Adattartalom változtatás

A képernyőn levő utolsó cim tartalmának megváltoztatását lehet elérni a / parancssal.
Alakja:

/

A / billentyü lenyomása után a tartalom megváltoztatása a cim megváltoztatásánál elmondott módon történik.

4.3. Listázás kijelölése

Ennek segitségével mondhatjuk meg, hogy a listázás hol történjen: printeren vagy képernyőn.

4.3.1. Listázás nyomtatón

A nyomtatón való listázási parancsa]
Alakja:

]

Hatására a disassemblálás a nyomtatóra történik.

4.3.2. Listázás képernyőn

Az utasitás a] , tehát megegyezik a 4.1.-el.

4.4. Memória kijelölése

A rutinokkal lehetséges a gép illetve a mágneslemez berendezés memóriatartalmának kiiratása, megváltoztatása.

Gép memória kiirása

A gép memóriájának (ROM, RAM) disassemblálását végzi a > parancs.

Alakja:

>

A floppy egység memóriája (ROM, RAM) vizsgálatát irja elő a < parancs.

Alakja:

<

4.5. Listázás vezérlése

A listázás vezérlésére egy sor billentyűt használhatunk. Ezek a következők:

RETURN	a cimet eggyel növeli
↑	cimet eggyel csökkenti
—	a beállitott cimtől folytatódik a disassemblálás, ha a tartalom értelmezhetetlen * * jelenik meg
space	folyamatos disassemblálás
CTRL	kiirás lassitása
STOP	kiirás megállitása
SHIFT	kiirás átmeneti megállitása
SHIFT/LOCK	benyomva tartásáig átmenetileg áll, elengedve tovább-

4.6. További lehetőségek

4.6.1. Gépi kódu program futtatása

A * billentyűvel indithatunk el egy gépi kódu programot. Ha a futás közben egy RST utasitást talál a gép, akkor visszatér monitor állapotba.

Ha a disk tároló van kijelölve (∠ parancs!), akkor akár a lemezegység tárolójában levő program is lefuttatható.

4.6.2. Memóriaterület másolása

Memóriaterületek máshova másolására szolgál a @ parancs. Ezzel mind ROM, mind RAM területről másolhatunk.

Alakja:

@

A parancs kiadása előtt be kell állitani a másolandó és a másolási terület paramétereit.

A másolandó terület megadása :

22-23 byte kezdőcim beállitása
24-25 byte végcim + 1 beállitása.

Másolás helyének megadása

"+" billentyűvel beállitjuk az első byte cimét. Ezután nyomjuk meg a másolási parancsot jelentő billentyűt.

4.6.3. Disassembler kikapcsolása

A kikapcsolás az = billentyű megnyomásával történik. Ekkor a BASIC értelmező és a HELP + egyéb parancsok ismét rendelkezésünkre állnak.

5. DOS parancsok

A disk operációs rendszer parancsainak rövidítése lehetséges a HELP + segítségével. A továbbiakban ezeket ismertetjük.

5.1. Programbetöltés

a.) Alakja:

/ n

ahol n a lemezről betöltendő programnév.
Hatása: megegyezik a LOAD "n", 8 paranccsal.

b.) Alakja:

% n

Hatása: megegyezik a LOAD "n", 8, 1 paranccsal.

5.2. Programbetöltés és indítás

Alakja:

↑ n

Hatása: megegyezik a LOAD "n", 8 :RUN parancsokkal.

5.3. Programkiírás lemezre

Alakja:

← n

ahol n a kiirandó program neve.

Hatása: megegyezik a SAVE "n", 8 paranccsal.

Hiba esetén azonban kiirja a statusz kódot és a hiba nevét is.
Névazonosság esetén az

OVERWRITE? JA/NEIN

kérdés iródik ki.
A további végrehajtás a válaszunktól függő:

J esetén a lemezen levő programot felülirja,
N esetén befejezi a parancs végrehajtását.

5.4. Kiirt program ellenőrzése

Alakja:

← n

ahol n a kiirt program neve.
Hatása: megegyezik a VERIFY "n", 8 paranccsal.

5.5. Lemeztartalom kiirása

Alakja:

Ø

Hatása: megegyezik a LOAD "Ø" , 8 :LIST parancsokkal, de a gépben levő aktuális programot nem irja felül a lista.

5.6. Disk státusz kiiratása

Alakja:

@

Hatása: kiirja az utolsó disk átvitel státuszát.

5.7. Parancscsatornáról olvasás

Alakja:

>

Hatása: a 15-ös parancscsatornáról egy hexadecimális jegyet beolvas.

5.8. Parancs csatorna parancsok

Az M- utasitások kivételével minden 15-ös csatornára vonatkozó parancsot kiadhatunk a @ segitségével.

A @ parancs után a PRINT # 15 után irandó paramétereket kell csak megadni. Igy elmarad az OPEN, CLOSE és a PRINT parancs. Például az ALMA nevü programot az alábbi parancsokkal lehet törölni a lemezről:

```
OPEN 15, 8, 15
PRINT #15, "SO:ALMA"
CLOSE 15
```

Ehelyett elegendő a következőket irni:

```
@ SO:ALMA
```

6. Assembler fordító

A HELP PLUS egy kétfázisu assembler fordítót is tartalmaz. A fordító a] utasítással aktivizálható.

Alakja:

]

A fordításhoz elő kell állitani lemezen vagy szalagon a forrásprogramot. Ezen program alakja eltér a standard assembler programok alakjától, mivel a fordító igényli az utasítások beszámozását.

A forrásprogram a következő alaku:

s (c) opc (op) (m)

ahol	s	az utasitás sorszáma (0 és 63999 közötti egész szám)
	c	nem kötelező paraméter, utasitáscimke
	opc	assembler mûvelet kódja
	op	nem kötelező, az utasitás operandusa
	m	nem kötelező paraméter, megjegyzés.

Megjegyzések:

1. Az utasitások mindegyike külön sorban kell legyen.
2. Kötelező utasitásonként: sorszám, utasitás kód.
3. A forditás két fázisban történik:

   I. fázis: cimke táblázat előállitása
   II. fázis: gépi kód előállitása

   Ennek megfelelően a lemezről kétszer olvasódik be a forrásprogram.

SIMON'S BASIC

Bevezetés

A SIMONS' BASIC egy a C64 BASIC értelmező kiterjesztése. Egy sor utasitást és parancsot tartalmaz a beépitett értelmezőhöz képest. A parancsok legtöbbjét programba beépitve, utasitásként is használhatjuk. Két verzióba létezik:

a) Cartridge forma
b) Floppy lemezen

Az a) esetben a rendszer aktivizálása a következőképpen történik:

a) kikapcsoljuk a C64 gépet, majd behelyezzük a Cartridge-t a gép hátoldali csatlakozójába,
b) a lemezről

LOAD "x ", 8

parranccsal betöltjük a SIMONS' BASIC-et betöltő programot,
a READY üzenet megjelenése után lefuttatjuk a gépben lévő betöltőt, azaz kiadjuk a

RUN/RETURN

parancsokat.

A lemezről ezután elindul a betöltés. A betöltés után a következő üzenetet irja ki a gép fekete betükkel fehér képernyőre:

x x x EXPANDED CBM V2 BASIC x x x
30719 BASIC BYTES FREE

Ezután rendelkezésünkre áll a több mint 100 uj utasitást és parancsot tartalmazó kibővitett verzió.

Megjegyzendő, hogy a rendelkezésünkre álló szabad terület mintegy 8K -val csökken. Ez az ára a kibővitett lehetőségek - nek.

1.1. Grafikus utasitások

A képernyőre rajzolás kényelmesebbé tétele céljából a SIMON BASIC egy sor utasitást tartalmaz. A fejezetben a rajzoló eljárásokat foglaljuk össze.

1.1.1. Rajzoló utasitások

Emlékeztetőül megjegyezzük, hogy a képernyő fekete/fehér módban 320x200, szines módban 160x200 elemi területre (cellára) van felosztva. Ezt onnan kapjuk, hogy fekete/fehérben egy karakter - hely 8x8 elemi helyre, szinesben pedig 8x4 helyre van felosztva. Az egyszerüség kedvéért a továbbiakban az alábbi jelölést használjuk:

HR	fekete/fehér grafikus mód (HIRES)
MC	szines grafikus mód (MULTI COLOUR)

Megjegyezzük, hogy a képernyőt ugy kell elképzelni, mintha egy koordináta rendszer IV. negyedében lenne. Különbség csak annyi, hogy y értéke nem negativ.
A képernyő bal felső sarka a (0,0) pont, a jobb felső sarka a (320,0) pont, a bal alsó sarka a (0,200) pont és végül a jobb alsó sarka a (320,200) pont.

A rajzoló utasitások mindegyikében meg kell adni a rajzolási tipust. Ez HR-ben és MC-ben különböző jelentéssel bir. A továbbiakban ezen tipusok megadásával foglalkozunk.

HR mód

tipus	jelentése
0	kitörli a cellát
1	megjeleniti a cellát
2	az eredeti cellaállapotot ellenkezőjére változtatja Értéke 0, ha 1 volt 1, ha 0 volt

MC mód

tipus	jelentése
0	kitörli a cellát
1	1-es szinnel megjeleniti a cellát MULTI/LOW COL parancsnál
2	2-es szinnel megjeleniti a cellát MULTI/LOW COL parancsnál
3	3-as szinnel megjeleniti a cellát MULTI/LOW COL parancsnál
4	a cella szinét inverzére változtatja a következőképpen:

0 -ról	3 -ra
1-es szinről	2 -re
2-es szinről	1 -re

1.1.2. Átkapcsolás grafikus módra

A HR módra való átváltás, az alap és háttérszin megadás a HIRES utasitással történik.

Alakja:

HIRES x, y

ahol:	HIRES	az átváltás kulcsszava,
	x	a rajzolás szine (az elemi cellák szine),
	y	a 8x8-as karakterhely szine (az un. háttérszin).

A megengedett szinek a C64 szinei, azaz

0	fekete
1	fehér
2	piros
3	türkiz
4	bibor
5	zöld
6	kék
7	sárga
8	narancs
9	barna
10	világos piros
11	1.szürke
12	2.szürke
13	világos zöld
14	világos kék
15	3.szürke

Példa: váltsuk át a gépet HR módba kék alapszinnel és narancssárga rajzoló szinnel.

Megoldás:

1.1.3. Átkapcsolás szines grafikus módra

Szines grafikus módra való átkapcsolást a MULTI utasitással érjük el.
Alakja:

MULTI x, y, z

ahol x, y, z a multikolor szinek.

Az utasitás előtt mindig egy HIRES utasitásnak kell állni. Az x, y, z szinekre érvényes rajzolási tipusnál elmondott paraméter - értéksor (lásd rajzoló utasitások paragrafus).

Példa: Legyen az első szin fehér, a második barna, a harmadik kék.

Megoldás:

HIRES 1,2:MULTI 1,9,6

Mint a fentiekben is látjuk, összesen 3 rajzolási szin adható meg a MULTI utasitással.
Ezt bőviti további 3 rajzolási szin lehetőséggel a LOW COL utasitás.
Alakja:

LOW COL p,q,r

ahol p,q,r a 0-15 értékek valamelyike.
Hatása: a MULTI-ban megadott x,y,z szinek helyett a gép a továbbiakban a p,q,r szinekkel dolgozik.

Az eredeti x,y,z szinekre való visszatérést a HI COL utasitással tudjuk elérni.
Alakja:

HI COL

Hatása: a továbbiakban a gép a MULTI utasitásban megadott szinekkel dolgozik.

Megjegyzés: egy programon belül több LOW COL utasitás is kiadható, igy tulajdonképpen az összes szint felhasználhatjuk rajzolásra.

1.1.4. Geometriai alakzatok megjelenitése

1.1.4.1. Egy pont megjelenitése

A PLOT grafikus utasitással lehetséges egy pont kirajzolása a képernyőn.

Alakja:

PLOT x,y,p

ahol x,y a kirajzolandó pont koordinátái,
p a rajzolási tipus.

Példa: rajzoljunk ki egy egyenest a képernyőn a PLOT utasitás felhasználásával.

Megoldás:

```
10   HIRES 1,2
20   FOR I=1 TO 320
30   PLOT I, O.5* I,1
40   NEXT I
50   GET A$ : IF A$ = "" THEN 50
```

A következő program fehér "homokszemekkel" szórja be a képernyőt:

```
10   HIRES 1, 2
20   FOR I=1 TO 200
30   FOR J=1 TO 320 STEP 10
```

```
40   PLOT RND (1) * J,RND (1) * I,1
50   NEXT J
60   NEXT I
```

1.1.4.2. Egyenes rajzolása

Egy egyenest a kezdő és végpont koordinátáinak megadásával, valamint a rajzolási tipus megadásával jelenithetünk meg a képernyőn.

Az egyenest rajzoló utasitás

```
LINE x1, y1, x2, y2, p
```

ahol	LINE	az egyenest rajzoló utasitás kulcsszava,
	x1,y1	az egyenes kezdőpontjának koordinátái,
	x2,y2	az egyenes végpontjának koordinátái,
	p	a rajzolási tipus.

Hatására az x1, y1 pontot összeköti az x2, y2 ponttal a megadott rajzolási tipus szerint.

Példa: rajzoljunk néhány egyenes vonalat a képernyőre.

Megoldás:

```
10   HIRES 0,2
20   LINE  30,0,0,200,1
30   LINE 310,0, 310, 200, 1
40   LINE  0,0,320,200,1
50   GO TO 50
```

A programot lefuttatva megkapjuk a kivánt eredményt.

1.1.4.3. Téglalap rajzolása

A REC utasitással egy téglalapot rajzoltathatunk ki a képernyőre. Alakja:

REC x1, y1, x2, y2, p

ahol REC — a téglalap rajzoló utasitás kulcsszava,
x1,y1 — a kirajzolandó téglalap bal felső sarkának koordinátái,
x2 — a téglalap alapjának hossza,
y2 — a téglalap magassága,
p — rajzolási tipus.

Példa: rajzoljunk ki egy négyzetet, amelynek bal felső sarkának koordinátái 10, 10, a négyzet oldala legyen 50 egységnyi, majd rajzoljunk ki egy ugyanezen kezdőponttól kiinduló 200 egység alapélü és 100 egység magasságu téglalapot.

Megoldáa:

```
10  HIRES 1,2
20  REC 10, 10, 50, 50, 1
30  REC 10, 10, 200, 100, 1
40  GET A$ : IFA$ = ""THEN 40
```

A következő program a REC utasitás használatának egy érdekes példája:

```
10  HIRES 1,2
20  FOR I=1 TO 320 STEP 10
30  FOR J=1 TO 200 STEP 10
40  REC I,J,I,J,1
50  NEXT J
60  NEXT I
```

A kirajzolás mechanizmusát igazán multikolor módban követhetjük végig.

Ehhez az alábbi módositásokat tegyük be a fenti programunkba:

```
15   MULTI 5,4,2
40   REC  I,J,I,J,RND (1) * 4
```

1.1.4.4. Görbevonalu zárt alakzat rajzolása

Lehetőség van szabályos görbevonalu alakzat (kör, ellipszis) rajzolására is a CIRCLE utasitással.

Formája a következő:

CIRCLE x1, y1, x2, y2, p

ahol	CIRCLE	a rajzoló utasitás parancsszava,
	x1, y1	az alakzat középpontja,
	x2	az alakzat "szélessége", vizszintes "sugár"
	y2	az alakzat "magassága", függőleges "sugár"
	p	a rajzolási tipus.

Mivel a képernyő nem négyzetes a rácspontok szempontjából, ezért kör rajzolása esetén az alábbi "torzitást" kell elvégeznünk:

HR mód	x2 = y2*1.3
MC mód	x2 = y2*1.6

Ha a kört ki akarjuk nyomtatni, ugy erre a torzitásra természetesen nincs szükség.

Példa:

```
10  HIRES 0,2
20  CIRCLE 100, 100, 100, 20, 1
30  GOTO 30
```

Egy másik alább közölt program véletlenszám generátor felhasználásával rajzol ki érdekes alakzatokat:

```
10  HIRES 2,5
20  A=INT (RND(1) *20)
30  B=INT (RND (1) *20)
40  C=INT (RND (1) *80)
50  D=INT (RND (1) * 90)
60  CIRCLE 120+A, 80+B, C, D, 1
70  GO TO 70
```

A CIRCLE utasitással kirajzolható alakzat egy részének kirajzolását végzi az ARC utasitás.

Alakja:

```
ARC x1, y1, a, b, c, x2, y2, p
```

ahol x1,y1	az alakzat középpontjának koordinátái,
a	a rajzolás "kezdetének" szöge,
b	a rajzolás "végének" szöge,
c	a lépésköz,
x2,y2	a CIRCLE utasitásnál elmondott paraméterek,
p	rajzolási tipus.

Hatása: x1, y1 középpontu alakzatot <u>a</u> szögtől a <u>b</u> szögig kirajzol c lépésközzel.

Megjegyzendő, hogy a és b fokban adandó meg és az alakzat függőleges átmérője az x1, y1 középponttól számitott egyenessel bezárt szöget kell megadni.

Példa: rajzoljunk ki egy 50,50 középpontu és 50,50 sugaru kört, majd ugyanennek a körnek "első negyedbe" eső cikkejét a 150,150 középponttal.

Megoldás:

```
10  HIRES 1,2
20  CIRCLE 50,50,50,50,1
30  ARC  150,150,0,90,1,50,50,1
40  GET  A$ : IF A$ = "" THEN 40
```

A lépésköz változtatására és az utasitás alaposabb megértésére futtassuk le a következő programot:

```
10  HIRES 1,2
20  FOR  I=10  TO 300  STEP 10
30  ARC 150,100,0,270,I,150,100,1
40  NEXT  I
50  GO TO 50
```

A CIRCLE utasitással kirajzolt alakzat hurjainak megrajzolását végzi az ANGL utasitás.

Alakja:

ANGL x1, y1, a, x2, y2, p

ahol x1, y1 — az alakzat középpontjának koordinátái,

a — a függőleges "sugárral" bezárt szög,

Hatása: Az x1, y1 középpontu, x2, y2 tengelyü ellipszisbe egy olyan egyenes rajzol, amely a szöget zárt be a függőleges sugárral.

Példa: rajzoljuk ki egy kör sugarait 30°-onként.

Megoldás:

```
10  HIRES 1,1
20  CIRCLE 160, 100, 100, 100, 1
30  FOR I=1  TO 360  STEP  30
40  ANGL 160, 100, I, 100,100,1
50  NEXT  I
60  GO  TO  60
```

A fenti programot módositsuk és futtassuk le az igy kapott programot. Cseréljük ki a következő utasitásokat:

```
20  CIRCLE 160,100,160,100,1
40  ANGL 160,100,I,160,100,1
```

A multi kolor változathoz készitsük el az alábbi programot:

```
10  HIRES 1,  2
20  MULTI  2,  6,  14
30  CIRCLE  80,100,80,100,1
40  PAINT 80, 100,  3
50  FOR  I=1  TO  360  STEP  10
60  ANGL 80,100,I,80,100,2
70  NEXT  I
80  GO  TO  80
```

1.1.4.5. Általános alakzat kirajzolása

A DRAW utasitással a "pont kurzort" mozgathatjuk vonalhuzással vagy anélkül.

Alakja:

DRAW "nnn...9", x,y,p

ahol	x,y	a "kurzor" mozgatás kezdete,
	nn..	az alább ismertetett számok valamelyike,
	p	rajzolási tipus.

Az n értékeit és hatását az alábbiakban összegezzük:

n értéke	müvelet
0	egy cellával jobbra mozgatás
1	egy cellával feljebb lépés
2	egy cellával lejjebb lépés
3	egy cellával balra lépés
4	megegyezik a 2-vel
5	egy cellával jobbra mozgatás rajzolással
6	egy cellával feljebb mozgatás rajzolással
7	egy cellával lejjebb mozgatás rajzolással
8	egy cellával balra mozgatás rajzolással
9	mozgatás befejezése.

Hatása: x, y koordinátáktól kezdve az idézőjelben megadott számoknak megfelelő mozgásokat (és rajzolásokat, ha vannak) hajtja végre.

Megjegyzések:

1. Az 1-4 számok csak mozgatják a "kurzort", mig az 5-8 rajzolnak is a képernyőre.

2. A DRAW utasitás csak a ROT utasitás után rajzol ki a képernyőre.

A DRAW paranccsal létrehozott alakzatokat a ROT utasitással jelenithetjük meg.

Alakja:

ROT s, m

ahol s — 1 és 8 közötti érték, a forgatás szöge,
m — az eredeti méret növelési faktora, 1 és 255 közötti érték.

Hatása: a DRAW utasitással megadott alakzatot az s értéknek megadott szögben m-szeresére növelve kirajzolja.

A forgatási szöget az alakzat eredeti helyzetéhez mérten kell érteni.

A szög (s) értékei:

s	forgatási szög
0	nincs forgatás
1	45°
2	90°
3	135°
4	180°
5	225°
6	270°
7	315°

Példa: Rajzoljunk ki egy A betűt az 50,50 ponttól kezdve, majd forgassuk meg növekvő méretekkel a 100, 100 pont körül.

Megoldás:

```
10   HIRES 1,0
20   B$ = "666666666666666555555"
           \_____________/\____/
                 15          6
30   C$="777777777777777111111111888888 9"
          \_____________/\_______/\____/
                15           9       6
40   ROT 0,1
50   DRAW B$+C$ , 50,50,1
60   FOR I=0 TO 7
70   ROT I,0.5*I+2
80   DRAW B$+C$ , 100, 100,1
90   NEXT I
100  GET A$:IF A$="" THEN GO TO 100
```

Ha lépésenként akarjuk végignézni a forgatást, akkor tegyük be a programba a következő utasitást:

```
85   GET A$:IF A$="" THEN 85
```

Ha a fenti programot szinesben is meg akarjuk csinálni, javasoljuk a következő módositások elvégzését:

```
15   MULTI 6,8,10

  --------------

80   DRAW B$+C$, 100, 100, (RND(1) * 3)+1
```

1.1.5. Alakzatok szinezése

1.1.5.1. Zárt alakzat szinezése

Egy zárt alakzat kiszinezése megoldható a SIMON BASIC-ban a PAINT utasitással.

Ennek formája a következő:

PAINT x,y,p

ahol PAINT — az utasitás kulcsszava,
x,y — az alakzat tetszésszerinti pontjának koordinátái,
p — rajzolási tipus.

Végrehajtása: a rajzolási tipusban megadott szinre kifesti a megadott alakzatot. Ha az alakzat nem volt zárt, akkor az egész képernyőt a megadott szinüre "festi" át.

Példa: szinezzünk ki egy ellipszist.

Megoldás:

```
10   HIRES 3,2
20   CIRCLE 110, 100, 130, 15,1
30   PAINT 110, 100, 1
40   GO TO 40
```

Az alakzatok szinezésére és a szinskála tanulmányozásához futtassuk le az alábbi programot:

```
10   HIRES 0,0
20   MULTI 1,1,1
30   FOR I=0 TO 15
```

```
40    REC 10*I, 0, 10, 200, 1
50    NEXT I
60    FOR K=0 TO 15
70    LOW COL K,K,K
80    PAINT 10*K+3, 100,1
90    NEXT K
100   GET A$:IF A$="" THEN GO TO 100
```

1.1.5.2. Szines téglalap rajzolása

Egy téglalapot a REC utasitással rajzolhatunk ki. Ezt szinezni a PAINT utasitással lehet. A két funkciót egyszerre és gyorsabban látja el a BLOCK utasitás.

Alakja:

```
BLOCK x1, y1, x2, y2, p
```

ahol

x1,y1,x2,y2	a REC utasitásnál elmondott paraméterek,
p	rajzolási tipus.

Hatása: kirajzol egy téglalapot az x1, y1, x2, y2 koordináták figyelembevételével a REC utasitásnál elmondottak szerint.

Amennyiben multi-color módot használunk, ugy a p az ott elmondott értékeket veheti fel.

Példa:

1. Rajzoljunk ki két négyzetet a képernyő jobb felső sarkában az utasitásokban megadott méretekben.

Megoldás:

```
10    HIRES 1,0
20    BLOCK 1, 1, 100, 100, 1
30    BLOCK 50, 60, 80, 100, 2
40    GET A$:IF A$="" THEN GO TO 40
```

Eredmény: képernyő bal felső sarkában egy fehér négyzet, majd egy fekete négyzet jelenik meg.

2. A fenti programot multi-color módban is elkészithetjük:

```
10  HIRES 1,0:MULTI 6,8,10
20  BLOCK 1,1,100,100,1
30  BLOCK 50, 60, 80, 100, 2
40  GET A$:IF A$="" THEN GO TO 40
```

Eredmény: Kék négyzeten belül sárga kis négyzet.

1.1.6. Rajzok feliratozása

1.1.6.1. Karakterek megjelenitése

A grafikus módban is szükséges bizonyos jelek, szövegek képernyőre irása. Egy karakter kiirását végzi el a CHAR utasitás.
Alakja:

CHAR x, y, c, p, s

ahol CHAR   a karakter rajzolás kulcsszava,
x,y   karakter kirajzolásának koordinátái,
c   a kirajzolandó karakter POKE kódja,
p   a rajzolási tipus,
s   a kirajzolandó karakter mérete.

Hatása: a megadott pontba kirajzolja a POKE kóddal megadott karaktert a rajzolási tipus szerint. Az s méret értéke 1, akkor normál méretben rajzolja, ha más, megfelelő méretalakitás történik.

Példa:

```
10  HIRES 12, 9
```

```
30   X=INT (RND(1) * 15)
40   CHAR T * 10, 80, T, 1, X
50   NEXT
60   GO TO 60
```

1.1.6.2. Karakterláncok kiiratása

Egyszerre több karaktert is megjelenitő utasitás a TEXT.
Alakja:

TEXT x,y, "(t) b", p, s, i

ahol	TEXT	a karaktersort megjelenitő utasitás kulcsszava,
	x,y	az első karakter koordinátái (szöveg kezdete!)
	t	az alsó illetve felső állást megadó paraméter,
	b	a kiirandó karaktersorozat,
	p	rajzolási tipus,
	s	a karakterek mérete,
	i	üres cellák száma a karakterek kezdete között.

A t értékei a következőképpen adhatók meg:

CTRL-a billentyü: nagybetük
CTRL-b billentyü: kisbetük.

Hatása: a megadott (x,y) ponttól kezdve megfelelő állásban vízszintesen kiirja a b szöveget a rajzolási tipusnak és a méretnek megfelelően.

A méretre vonatkozó előirásokat a CHAR utasitásnál mondtuk el.

Példa:

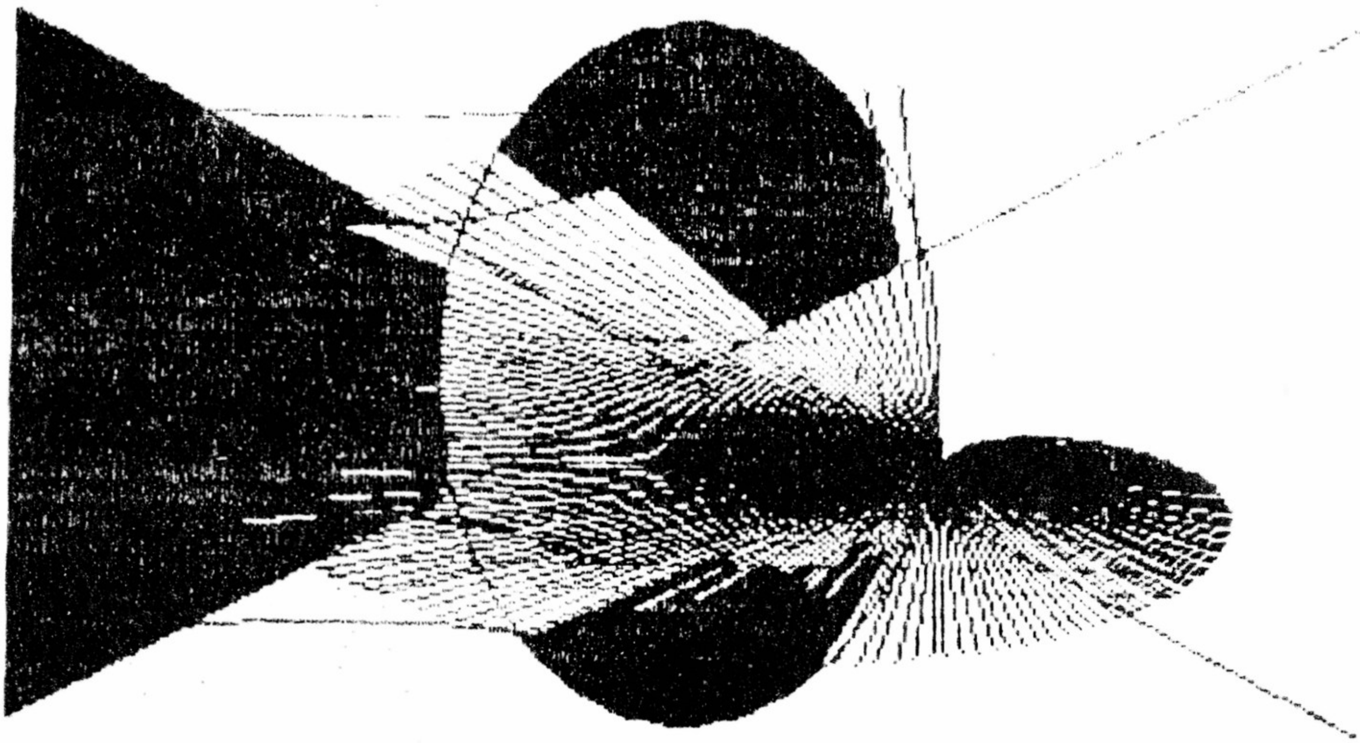
```
10   HIRES 6,4
20   TEXT 60, 30, "B IROK IS", 1,3,10
30   TEXT 60,60 "UGYE JO?", 1,5,20
40   TEXT 60,100 "A ALMAT ESZEM", 1,8,25
50   GO TO 50
```

1.1.6.3. Mintaprogram

Az alábbiakban több utasitást használó programot közlünk:

```
10   HIRES 2, 5,
20   LINE 0, 0,320,200,1
30   LINE 320,0,0,200,1
40   CIRCLE 160,100,60,100,1
50   ARC 80,0,152,2o7,1,100,30,1
60   ARC 80,200,333,28,1,100,30,1
70   ARC 80,0,142,217,1,100,60,1
80   ARC 80,200,323,38,1,100,60,1
90   FORT=5TO358STEP2
100  ANGL 220,125,T,T/1.5,T/3.3,1
110  NEXT
120  PAINT1,1,1
130  PAINT 200,100,1
140  PAINT 160,20,1
150  PAINT 160,180,1
160  COPY
170  GO TO 170
```

A fenti programhoz tartozó rajzot a nyomtatón készitettük el:



1.1.7. További grafikus utasitások

1.1.7.1. CSET utasitás

A CSET utasitással lehetőség van a karakterkészletek váltására és a legutolsó grafikus képernyő visszarajzolására is.

Alakja:

CSET i

ahol i — 0 és 2 közötti érték

Hatása: i = 0 esetben a gép felső állása/grafikus karakterkészletét választja

i = 1 esetben a felső/alsó állás karakterkészletét választja

i = 2 esetén átvált grafikus üzemmódba és kirajzolja a képernyőre a grafikus képernyő utolsó rajzát.

Megjegyzés: ha az utolsó rajz szines üzemmódban jelent meg, akkor a CSET 2 utasitást egy MULTI utasitásnak kell követni.

1.1.7.2. Egy pont állapotának lekérdezése

Sokszor szükséges a grafikus képernyőn egy "pontról" eldönteni, hogy történt-e oda rajzolás vagy sem. Ezt végzi el a TEST függvény.

Alakja:

TEST (x, y)

ahol x,y — a keresett pont paramétere

Hatása: - a TEST numerikus értékü függvény értéke 1, ha a pontban történt rajzolás,

- ha a "pont üres" akkor a TEST értéke 0.

1.2. Hagyományos utasitások kiterjesztése

1.2.1. Feltételes utasitás kiterjesztése

1.2.1.1. IF ... THEN ... ELSE

A BASIC-ban megadott IF...THEN szerkezet mellett az IF...THEN...ELSE kifejezést használhatjuk.
Az utasitás alakja:

$$\text{IF L THEN } u_1 [:u_2:\ldots:u_n] \text{ :ELSE:} v_1 [:v_2:\ldots:v_m]$$

ahol L logikai kifejezés (reláció) ,
u_1, v_1 végrehajtható BASIC utasitások.

Végrehajtása: ha L igaz, az $u_1 (u_2 \ldots, u_n)$ utasitások hajtódnak végre, egyébként a $v_1 (, v_2 \ldots, v_m)$ utasitások.

Példa:

IF 5 < a + b THEN X = 50:ELSE:X = 100
Ha 5 < a + b, akkor X értéke 50, egyébként X értéke 100.

1.2.1.2. RCOMP utasitás

Az RCOMP utasitás formája:

$$\text{RCOMP:} u_1 [:u_2 \ldots u_n] \text{ :ELSE:} v_1 [:v_2 \ldots :v_n]$$

ahol RCOMP, ELSE az utasitás kulcsszavai,
u_i, v_i végrehajtható utasitások.

Végrehajtása: ha az RCOMP utasitást megelőző IF utasitásban a feltétel értéke igaz, akkor az u_i, egyébként a v_1 utasitások hajtódnak végre.

Példa:

```
10  IF X+Y = Z THEN P = 20:ELSE:P=30
20  RCOMP : R = 15 : ELSE : R = 1000.25
```

Eredmény: ha X + Y = Z, akkor P = 20, R = 15, másküIönben P = 30, R = 1000.25

1.2.2. Ciklusutasitás kiterjesztése

A FOR ciklusutasitás mellett egy sor más ciklusutasitást is használhatunk a SIMON BASIC-ban.

1.2.2.1. REPEAT ... UNTIL utasitás

Ennek formája:

c_1 REPEAT
c_2 u_1
. . . .
c_n u_{n-1}
c_{n+1} UNTIL L

ahol REPEAT, UNTIL a ciklusutasitás cimszavai
c_i utasitások számai (i=1,2,...,n+1),
L logikai kifejezés (reláció),
u_i utasitássorozat.

Végrehajtása: a c_2 ... c_n utasitások addig kerülnek végrehajtásra, ameddig az L kifejezés értéke igaz nem lesz.

Példa:

```
100  REPEAT
110  INPUT X
120  UNTIL X = 50
```

A fenti utasitássorozat addig kerül végrehajtásra, ameddig az 50-es értéket be nem olvastuk az X-be.

1.2.2.2. LOOP ... EXIT IF ... END LOOP

Az utasitás formája a következő:

c_1 LOOP

c_2 u_1

.

.

c_i EXIT IF L

.

.

c_{n+1} END LOOP

ahol LOOP, EXIT IF, END LOOP az utasitás kulcsszavai

c_i az utasitások számai (i=1,2, ...,n),

L logikai kifejezés (reláció),

u_i utasitássorozat.

Végrehajtása: a $c_2 \ldots c_n$ utasitások addig hajtódnak végre, ameddig az L logikai kifejezés igaz nem lesz.

Példa:

```
100 LOOP
110 INPUT X
120 EXIT IF X = 50
130 END LOOP
```

A fenti utasitássorozat addig hajtódik végre, amedding az input adatok között az 50-es elő nem fordul.

Megjegyzés: a c_i utasitássorozatban az EXIT IF utasitás többször is előfordulhat.

1.2.3. Zárt szubrutin

Szubrutint a GOSUB utasitás mellett a PROC utasitással is létrehozhatunk. Az ilyen szubrutin un. zárt szubrutin lesz.

Formája:

c_1 PROC n

c_2 u_1

.

.

c_n u_{n-1}

c_{n+1} END PROC

ahol	PROC	eljárásfej,
	n	eljárásnév,
	c_i	i=2,...,n szubrutin törzsét alkotó eljárások,
	END PROC	szubrutin végét jelző utasitás.

A szubrutint a következő utasitással lehet végrehajtani:

EXEC n

ahol EXEC a szubrutint hivó utasitás kulcsszava,

n a kiirandó szubrutin neve.

Végrehajtása: az n nevű szubrutint végrehajtja, majd az EXEC utasitást követő utasitás után folytatódik a program végrehajtása.

Példa:

```
100  EXEC ABC
110  PRINT "SZUBRUTIN VÉG"
120  . . .
300  PROC ABC
310  FOR I=1 TO 100 : PRINT I, I*I
320  NEXT I
330  END PROC
```

A program végrehajtása során az EXEC aktivizálja az ABC nevű szubrutint. Igy a képernyőre 1-től 100-ig ezen számok és a négyzeteik kerülnek kiirásra. Ezután a SZUBRUTINVEG kiirással folytatódik a program végrehajtása (110-es utasitás!).

1.2.4. Cimkére ugrás

A PROC utasitást a zárt szubrutin létrehozása mellett ugró utasitás egy másik formájának alkalmazásához is használhatjuk.

Formája:

c PROC n

ahol	c	utasitáscimke,
	PROC	ugrás helyét megadó parancs,
	n	szimbolikus név.

Végrehajtása: a c cimkétől kezdődő utasitássorozatot a CALL n utasitással aktivizálhatjuk. Igy a CALL n végrehajtás szempontjából megegyezik egy GO TO c utasitással. A különbség csak az, hogy az n egy név, nem pedig egy cimke.

Jegyezzük meg: zárt szubrutinra nem alkalmazhatjuk a CALL utasitást.

Példa:

```
...........
100  CALL  ALMA
110  X = 25
...........
200  PROC  ALMA
210  X = 50
220  END  PROC
```

A fenti utasitássorozat hatására először X az 50, majd a 25 értéket veszi fel.

1.2.5. Helyi és globális változók

Sokszor szükséges a programirás során azonos változókat más célra használni. Különösen fontos ez, ha többen dolgoznak egy program kifejlesztésén és azonos változókat használnak. Valamely programrészben (különösen szubrutinoknál) egy változó értékének elmentését, és uj változóként való használatát a LOCAL utasitással tudjuk megoldani. Mindaddig érvényes az utasitás, ameddig egy GLOBAL parancsot ki nem adunk, aminek hatására a változó aktuális értékei a LOCAL parancs előtti értékek lesznek.

A parancsok formái:

$$c \quad \text{LOCAL} \quad v_1, v_2, \ldots, v_n$$

$$c \quad \text{GLOBAL} \quad w_1, w_2, \ldots, w_m$$

ahol c utasitáscimke,
LOCAL, GLOBAL utasitás kulcsszók,
v_i, w_i változó nevek.

Végrehajtásuk: a LOCAL parancs hatására a $v_1, v_2, \ldots, v_n$ változók értéke eltárolódik és értékük nulla lesz.

Ezen változók értéke a rájuk vonatkozó utasitásig a helyi használat eredményeképpen kapott érték lesz. A GLOBAL utasitás után a w_1 listában szereplő változók visszakapják a rájuk vonatkozó LOCAL parancs előtti értéküket.

Példa:

```
. . . . . . . . . . . .
100  X=425.2:  Y$="ALMA" :Z%=11111
110  PRINT  X,Y$,Z%
120  LOCAL  X,Y$
130  PRINT  X,Y$,Z%
140  X=1.45:Y$="HOLVOLT"
150  PRINT  X,Y$,Z%
160  GLOBAL  Y$:LOCAL Z%:Z%=22222
170  PRINT  X,Y$,Z%
180  GLOBAL  X
190  PRINT  X,Y$,Z%
```

A fenti utasitások hatására a következők jelennek meg a képernyőn:

```
425.2   ALMA       11111
  0      11111       0
  1.45   HOLVOLT 11111
  1.45  ALMA        22222
425.2   ALMA       22222
```

1.2.6. Számitott ugró utasitás

Aritmetikai kifejezés értékétől függő utasitás a CGOTO.

Alakja:

, CGOTO e

ahol e aritmetikai kifejezés.

Hatása: GO TO c utasitással egyezik meg, ahol c az e kifejezés értéke.

Megjegyzés: Amennyiben e valós, akkor a c az e értékének

1.2.7. DATA pointer állitása

A READ utasitás a BASIC-ben ugy hajtódik végre, hogy beolvassa a DATA aktuális pozíciójától a kivánt adatokat. Az aktuális pozició megváltoztatására szolgál a RESET utasitás.

Alakja:

RESET c

ahol c egy DATA utasitás cimkéje.

Hatása: a következő READ utasitás a c cimkével megadott DATA utasitásra vonatkozik.

Példa:

```
10  INPUT NØ
20  IF NØ = "A"  THEN  RESET 80:GO TO 40
30  RESET 90
40  FOR I=1 TO 10
50  READ A:PRINT A
60  NEXT I
70  STOP
80  DATA 1,2,3,4,5,6,7,8,9,10
90  DATA 10,20,30,40,50,60,70,80,90,1,2
```

A program végrehajtása során

- ha az A billentyüt nyomjuk meg, akkor a gép kiirja az 1,2,...,9,10 adatokat

- ha a B billentyüt nyomjuk meg, akkor a 10,20,...,90,1 adatokat.

1.2.8. Olvasással-írással kapcsolatos müveletek

1.2.8.1. Szöveg kiirás középre

Képernyőre szövegek elhelyezésére szolgáló utasitás a CENTRE.
Alakja:

CENTRE "s"

ahol s a képernyőre elhelyezendő karakterlánc.

Hatása: az s karakterláncot elhelyezi a képernyő aktuális sorának közepére.

Példa: A képernyő aktuális sorának közepére irassuk ki a

VEGEREDMENY

szöveget.

CENTRE "VEGEREDMÉNY"

1.2.8.2. Kiiratási kép szerkesztése

Numerikus adatok kiiratási képének meghatározására szolgál a USE...PRINT utasitás.
Alakja:

a) USE "p.q",v:PRINT
b) USE "s1 p.q s2",v:PRINT

ahol p,q # jelek sorozata,
s1,s2 karakterláncok,
v karakter tipusu változó.

Hatása:

a) a v változóban levő numerikus értéket p egészre és q tizedesre kiirja a képernyőre.

b) A v változóban levő numerikus értéket az az a)-ban leirtak szerint kiirja, de előtte az s1, utána az s2 szöveget is megjeleniti.

Példa: Tekintsük a következő programot

```
10  INPUT A
20  A$=STR$(A)
30  USE "EGESZ:                TIZEDES",A$:PRINT
40  GO TO 10
```

A fenti programot futtassuk a következő 4 értékkel

0.5
125.589
2.41
1125.5643

A képernyőre a gép a következőket irja ki:

```
EGESZ:     .5 TIZEDES
EGESZ: 125.58TIZEDES
EGESZ:   2.41TIZEDES
EGESZ: 125.56TIZEDES
```

Megjegyzendő, hogy a kiiratás során - hibajelzés nélkül - kezdő és végszámjegyek tünhetnek el, mint az az utolsó érték kiiratásánál is előfordult. Ezért a USE utasitás használata esetén ügyeljünk a számok nagyságrendjére.

1.2.8.3. Kiiratás helyének megadása

Szövegek adott sor és oszlop kezdettel történő kiiratását végzi a PRINT AT utasitás.

Alakja:

a) PRINT AT(p,q)"s"

b) PRINT AT(p,q)v

ahol s karakterlánc,

v változó,

p,q oszlop illetve sorszám.

Hatása: a) a p. oszlop q. sorától kezdve kiirja az s szöveget

b) a p. oszlop q. sorától kezdve kiirja a v változó értékét.

Megjegyzendő, hogy több AT összetétel is előfordulhat egy PRINT utasitásban.

Példa: Irassuk ki az OSSZESEN szöveget a 3. sor 15. oszlopától az ALAPBER szöveget az 5. sor 10. oszlopától és a POTLEKOK OSSZEGE szöveget ugyanezen sor 20. pozi ciójától.

Megoldás:

```
10  PRINT AT(15,3) "OSSZESEN" AT(10,5)
    "ALAPBER"
20  PRINT AT(20,5)"POTLEKOK  OSSZEGE"
```

Megjegyzendő, hogy a sorok 0-tól 24-ig, az oszlopok 0-tól 39-ig számozódnak.

1.2.8.4. Bemenő adatok ellenőrzése

A bemenő adatok beolvasására, tipusának és maximális hosszának ellenőrzésére használatos a FETCH utasitás.

Alakja:

FETCH "v", n, vn

ahol v az utasitás vezérlő karaktere,
n a beolvasandó adat maximális hossza karakterekben,
vn a változó neve, ahova a beolvasás történik.

Hatása: a vn változóba beolvas a billentyüzetről legfeljebb n karaktert a v vezérlő karakternek megfelelően.

A v értékei a következők lehetnek:

karakter értéke	megengedett input karakter
CLR/HOME	nem siftelt alfabetikus karakter
CURSOR DOWN	numerikus karakter
CURSOR LEFT	alfabetikus és siftelt karakter

Ha a karakterlánc tipusa nem megfelelő, akkor a futás nem folytatódik, amig megfelelő tipusu adatot nem kapott. Megengedett hosszabb lánc esetén csak az első n számot fogadja el.

Példa:

```
10   FETCH "CLEAR/HOME",2,A$
20   PRINT "TOVABB"
. . . . . . . . .
```

A fenti program csak akkor indul tovább, ha egy, vagy két betüt leütünk és megnyomjuk a RETURN gombot.

1.2.8.5. Funkció billentyü használata

Az F1-F8 billentyük egyik felhasználását segiti az INKEY függvény.

Alakja:

INKEY

Hatása: - az INKEY értéke nulla, ha egyik billentyüt sem nyomtuk le,

- és 0,..,16 ha F1,...,F4 és esetleg a SHIFT vagy COMM. billentyü valamelyikét nyomjuk le

Példa: Tekintsük az alábbi programot:

```
10  A=INKEY
20  IF A=0 THEN 10
30  PRINT A
40  ON A GO TO
      100,200,300,400,500,600,700,800
. . . . . . . . .
```

A fenti program addig "olvassa" a funkció billentyüket, ameddig valamelyiket le nem nyomtuk. Valamely funkció billentyü lenyomása után a megfelelő százassal kezdődő cimkére ugrik.

Példa: tekintsük a következő programrészt:

```
10  PRINT "NYOMJA MEG AZ F1 BILLENTYUT"
20  INKEY
30  IF B=1 THEN 60
40  PRINT "NEM F1-ET NYOMOTT"
50  GO TO 10
```

A fenti programot futtatva a végrehajtás csak akkor folytatódik a 60-as utasitásnál, ha az F1 billentyűt megnyomtuk.

Megjegyzés:

Az INKEY értékei

Billentyű nyomás	INKEY
F1	1
SHIFT, F1	2
F3	3
SHIFT, F3	4
F5	5
SHIFT, F5	6
F7	7
SHIFT, F7	8

1.3. Egyéb utasitások

A paragrafusban egy sor hasznos utasitást mutatunk be. Ezek közül kiemeljük az egyik leghasznosabbat, amellyel az előforduló hibákat a programból "kijavithatjuk" és a futás igy tovább folytatható. Az utasitás megfelelőjét a PL/I-ből a nyelvet ismerők már használhatták.

A programok titkositásával az illetéktelen továbbfejlesztők, programrészeket egyéb célokra felhasználók munkáját lehet megneheziteni.

1.3.1. Programfutás felfüggesztése

A program futásának megadott időre történő felfüggesztését végzi a PAUSE utasitás.

Alakja:

a) PAUSE "u",i

b) PAUSE i

ahol u tetszésszerinti szövegkonstans,

i másodpercek száma.

Hatása: a) A PAUSE utasitáshoz érve a futást felfüggeszti i másodpercre és kiirja az idézőjelbe irt szövegkonstanst.

b) A futás felfüggesztődik i másodpercre.

A program végrehajtása minskét esetben automatikusan folytatódik i másodperc mulva. Ha az i másodpercet nem akarjuk kivárni, akkor a RETURN billentyűvel továbbithatjuk.

Példa:

```
10   PAUSE "ADD MEG AZ A-t", 30
20   INPUT A
. . . . .
```

A fenti programrész végrehajtása során a gép kiirja az ADD MEG AZ A-t szöveget, 30 másodpercig áll, majd végrehajtja a 20-as utasitást.

1.3.2. Hibák lekezelése programból

A futás közben előforduló hibák egy részét lekezelhetjük programból is, igy ezek miatt nem szakad meg a program futása.
A következő hibák lekezelése lehetséges igy:

1 tul sok file

2 állomány már meg van nyitva

3 állomány nincs nyitva

4	állományt nem találja
5	egység nem áll rendelkezésre
10	NEXT FOR nélkül
11	szintaktikus hiba
12	RETURN GOSUB nélkül
13	DATA utasitásban már nincs adat
14	nem megengedett mennyiség
15	tulcsordulás
16	memóriát tulhaladta
17	ilyen utasitáscimke nem létezik
18	nem megfelelő tömbelem megadás
19	tömböt ujra definiálunk
20	nullával való osztás
21	csak direkt módban használható
22	tipus keveredés
23	tul hosszu karaktertipusu konstans

A fenti hibák esetén a hibaanalizálást a következő változók segitik elő:

ERRLN	hibás utasitás sorszáma
ERRN	hiba kódja

A hiba kezelését a következő utasitással lehet megoldani:

ON ERROR:GO TO c

ahol ON ERROR a hibakezelés kulcsszava,
c utasitáscimke, amelyre a hiba esetén ugrik.

A hiba lekezelését a fenti két (ERRLN, FRRN) változók felhasználásával végezhetjük el.

A tennivalók elvégzése után a

NO ERROR

paranccsal továbbindithatjuk a futást.
Az OUT parancs adja vissza a vezérlést a COMMODORE-64 BASIC hiba kezelő rutinjának.

1.3.3. Programok titkositása

Egy program mások által történő átalakitásának meggátolására a program megjelölt sorait el lehet "tüntetni". Az eltüntetendő sorok megjelölésére szolgál a DISAPA utasitás, sorok eltüntetését végzi a SECURE parancs.

A DISAPA utasitás formája:

DISAPA:

Ezt a formát kell beütni azon utasitások sorszáma után, amelyeket el akarunk tüntetni.

Példa: Tekintsük az alábbi programot:

```
10  DIM A(100,20)
20  FOR I=1 TO 100
30  FOR J=1 TO 5
40  IF I=J THEN A(I,J) = 999
50  NEXT J
60  NEXT I
```

Ha a program 20. és 40. sorszáma utasitásait el akarjuk tüntetni, akkor megjelöljük a DISAPA utasitással, azaz kicseréljük a következőkre:

```
20   DISAPA:FOR I=1 TO 100
40   DISAPA:IF I=J THEN A(I,J)=999
```

Ezután tüntessük el a kijelölt utasitásokat, azaz adjuk ki a SECURE 0 parancsot:

```
SECURE 0/RETURN
```

Listázzuk ki a programot (LIST RETURN).
Az eredmény:

```
10   DIM A(100,20)
20
30   FOR J=1 TO 20
40
50   NEXT J
60   NEXT I
```

2. BASIC függvények kiterjesztése

2.1. Matematikai függvények

2.1.1. Modulus függvény

A függvény formája a következő:

MOD (a,b)

ahol MOD a modulus függvény kulcsszava,
a,b aritmetikai kifejezés.

Értelmezése: a függvény értéke a modulo b.

2.1.2. Egészrész

A függvény formája a következő:

DIV (a,b)

ahol DIV az egész osztás függvény kulcsszava,
a,b aritmetikai kifejezés.

Értelmezése: a függvény értéke az a legnagyobb egész szám, amelyet b-vel megszorozva a-tól kisebb vagy egyenlő számot kapunk.

Példa: Tekintsük a következő programot:

```
10  INPUT A,B
20  PRINT A, ":", B, "=", DIV (A,B)
30  PRINT "A  MARADEK", MOD(A,B)
```

Lefuttatva a következők iródnak ki a képernyőre:

A	B	KIIRÁS
21	6	A:B = 3
		A MARADEK 3
16	2	A:B = 8
		A MARADEK 0

2.1.3. Tizedesrész függvény

A függvény formája:

FRAC (k)

ahol FRAC a függvény kulcsszava,
k aritmetikai kifejezés (konstans, változó...).

Végrehajtása: A függvény értéke a k kifejezés tizedes része.

Példa: a) PRINT FRAC (3.456), FRAC (1/2)
A képernyőn a következő számok iródnak ki: .456, .5

b) X = FRAC (12/5)
Az X változó értéke 0.4 lesz.

2.2. Logikai müveletek kiterjesztése

A BASIC-ben meglevő AND, OR és NOT logikai operátorok mellett a kizáró vagy müveletet is használhatjuk a SIMON BASIC-ben.
A logikai függvény alakja:

EXOR (a, b)

ahol EXOR a kizáró vagy függvény kulcsszava,
a,b egész (vagy valós) számok.

Végrehajtása: az a és b számokat bináris számmá alakitja, számjegyenként végignézi, ha a két számjegy azonos, akkor az eredmény 0, egyébként 1.

Példa:

a = %00101101
b = %00011100

eredmény 00110001 ezért a 49 érték iródik ki a

PRINT EXOR (%00101101, %00110001)

hatására.

2.3. Karakter müveletek

2.3.1. Beszurás karaktersorozatba

Sokszor szükséges egy karaktersorozatba egy másikat beszurni. Ezt végzi el az INSERT függvény.
Alakja:

INSERT ("s1", "s2", n)

ahol s1,s2 karakterláncok,
s1 a beszurandó karakterlánc,
s2 a karakterlánc, amelybe a beszurás történik,
n azon pozició helye, ahova a beszurás történik s2-ben.

Megjegyzendő, hogy s_1 és s_2 karaktereinek együttes száma nem haladhatja meg 255-öt.

Példa: Tekintsük a következő programot

```
10  A$ = "ALMA  ALATT"
20  B$ = INSERT ("A  FA",A$,5)
30  C$ = INSERT ("MEG  EGY", "EGY  KETTO",4)
40  PRINT  B$
50  PRINT  C$
```

A program lefuttatása után a következő szövegeket irja ki a gép a képernyőre

```
ALMA  A  FA  ALATT
EGY  MEG  EGY  KETTO
```

A függvényt karakterláncok összehasonlitása esetén is használhatjuk. Az összehasonlitás baloldalán egy numerikus tipusu változó van, amelynek értéke -1, ha az összehasonlitás igaz, egyébként 0.
Példa: Tekintsük a következő programot:

```
10  A = INSERT ("-OTT", "ITT",3) = "ITT-OTT"
20  PRINT  A
```

A futtatás eredményeképpen a következő érték jelenik meg a képernyőn:

```
-1
```

A függvénnyel kapcsolatos hibaüzenetet a Függelékben irjuk le.

2.3.2. Karakterlánc egy részének kicserélése

Egy karakterlánc végének a megváltoztatására szolgál az INST függvény.

Alakja:

INST ("s1", "s2", n)

ahol s1, s2 karakterláncok,
s2 a karakterlánc, aminek a végét változtatjuk meg,
s1 az s2 karakterlánc végére irandó karakterlánc,
n azon pozició helye az s2-ben, amelytől kezdve az s1 láncot helyettesitjük.

Az s1 és s2 akármelyike lehet karaktertipusu változó is.

Példa: Tekintsük a következő programrészletet:

```
10  AØ = INST ("ALMA", "PIROS-KEK",5)
20  PRINT AØ
```

Lefuttatva a következőket irja ki a gép:

PIROSALMA

2.3.3. Karakterlánc helyének megkeresése

Egy karaktersorozatban egy részsorozat megkeresésére és a helyének megállapitására szolgál a PLACE függvény.

Alakja:

PLACE ("s1", "s2")

ahol s1, s2 karakterláncok,
s1 a karakterlánc, amit keresünk,
s2 az a karakterlánc, amelyben az s1-et keressük.

A függvény értéke az s1 karakterlánc első poziciójának a helye s2-ben, ha s1 része s2-nek. Ellenkező esetben a függvény értéke nulla.

Példa: Tekintsük az alábbi programrészletet

```
10  A$="ALMA A FA ALATT"
20  X=PLACE ("FA", A$)
30  Y=PLACE ("ALA",A$) :Z=PLACE("N",A$)
40  PRINT X,Y,Z
```

A program végrehajtása során az alábbi értékek iródnak ki a képernyőre:

8 11 0

2.3.4. Karakterlánc többszörözése

Karakterlánc többszörözésére szolgáló eljárás a DUP.
Alakja:

DUP("s", n)

ahol s a sokszorozandó karakterlánc,
n egész szám, amely a többszörözések száma.

Hatása: az s karakterláncot n-szer irja egymásután.

Példa: Tekintsük az alábbi programsort

```
10  PRINT DUP ("XYZ3-SZOR",3)
```

A végrehajtás eredménye a következő kiirás:

XYZ3-SZORXYZ3-SZORXYZ3-SZOR

Végrehajtása: a h tizenhatos számrendszerbeli számot tizes számrendszerbeli számmá alakitja át.

Példák:

a) PRINT $00AB, $0C0A

Képernyőn a következő számok jelennek meg a végrehajtás után:

171 3082

b) X = $0CCC

Az utasitás végrehajtása után az X változó értéke:

3276

2.5. Fényceruzával, paddle-vel és a botkormánnyal kapcsolatos függvények

2.5.1. Fényceruza koordinátái

A fényceruza x koordinátáját adja meg finom grafikában a PENX függvény.

Alakja:

PENX

Értéke: a fényceruza x koordinátája (0 és 320 között).

Az y koordináta értékét veszi fel a PENY függvény.

Alakja:

PENY

Értéke: a fényceruza y koordinátája (0 és 199 közötti szám).

2.5.2. Paddle potenciométerének állása

A paddle ellenállását adja meg a POT függvény.
Alakja:

POT (i)

ahol i a berendezés száma (0 v. 1).
Értéke: a paddle potenciométerének állása, 0 és 255 közötti érték.

2.5.3. Botkormány helyzete

A botkormány állásának és a tüzelőgomb állapotának meghatározására alkalmas a JOY függvény.
Alakja:

JOY

Értékei:

128	tüzelőgomb meg van nyomva
0	botkormány alapállásban
1	botkormány fel
5	botkormány le
3	botkormány jobbra
7	botkormány balra
8	botkormány fel balra
2	botkormány fel jobbra
6	botkormány le balra
4	botkormány le jobbra

3. Képernyő müveletek

3.1. Képernyőn szin villogtatás

A képernyőn egy szin villogtatására szolgál a FLASH utasitás.
Alakja:

FLASH sz, s

ahol sz szinkód,
s sebesség.

Hatása: a képernyő aktuális szinét a megadott sz szinkóddal folyamatosan felcseréli a megadott sebességgel.
A sebesség 1 és 255 között változtatható.
Egy egységhez 1/16-od másodperc tartozik.

Megjegyzendő, hogy a FLESH utasitás a finom grafika esetén nem alkalmazható.
A FLASH utasitást, azaz a képernyő villogtatását az OFF utasitással hatástalanithatjuk. Az utasitásnak paramétere nincs, az aktuális FLASH utasitást hatástalanitja.

3.2. Keretszin villogtatása

A képernyő keretének szineit villogtatja a BFLASH utasitás.
Alakja:

BFLASH s, sz1, sz2

ahol s a sebesség,
sz1,sz2 szinkódok.

Hatása: az sz1, sz2 szinkóddal megadott szinek felváltva villognak a kereten az s sebesség szerint.

Az sz1 és sz2 kódok értéke 0 és 15 közötti egész számok lehetnek. Az s sebesség értéke 0 és 255 közötti egész szám. Egy egység 1/16-od másodperc. A BFLASH hatása kikapcsolható a

BFLASH 0

utasitással.

Példa: Futtassuk le az alábbi programot:

```
10  BFLASH 8,2,3
20  FOR I=1 TO 2000:NEXT I
30  BFLASH 0
```

Hatására a képernyő kerete piros/zöld szinekben villog, körülbelül három másodpercig.

3.3. Terület feltöltése karakterekkel

A képernyő egy részének megadott karakterrel való kitöltésére szolgál az FCHR utasitás.

Alakja:

FCHR a,b,c,d,e

ahol a 0,24 közötti érték, sorszám,
b 0,39 közötti érték, oszlopszám,
c a terület szélessége karakterekben,
d a terület mélysége karakterekben,
e a területen megjelenitendő karakter POKE kódja.

Hatása: az a. sor b. oszlopában c karakter szélességü és d karakter mélységü területen kirajzolja az e POKE kóddal megadott karaktert.

Példa: A képernyő jobb felső sarkában egy 20x20-as négyzetet töltünk fel Q betükkel.

Megvalósitás:

```
10 PRINT "SHIFT/CLR/HOME" :POKE 53281,5
20 FCHR 0, 0, 19, 19, 17
```

3.4. Egy terület szinének megválasztása

Egy területen megjelenő jelek szinének megváltoztatására szolgál az FCOL utasitás.

Alakja:

FCOL a,b,c,d,e

ahol a 0, 24 közötti érték, a terület kezdetének sorszáma,
b 0, 39 közötti érték, a terület kezdetének oszlopszáma,
c a terület szélessége karakterekben,
d a terület mélysége karakterekben,
e 0 és 15 közötti érték, a területen megjelenitendő karakterek szine.

Hatása: a területen megjelenő karakterek szine a megadott c szinü lesz.

Példa: A jobb felső 20x20 karakternyi területen megjelenő karakterek szine legyen piros.

Megoldás:

FCOL 0,0,19,19,2

3.5. Terület feltöltése szines karakterekkel

Egy megadott terület megadott szinü karakterekkel való feltöltésére szolgál a FILL utasitás.

Alakja:

FILL a,b,c,d,e,f

ahol az első 4 paraméter (a,b,c,d) a terület megadására szolgál (lásd FCHR, FCOL utasitások),

e a területen megjelenitendő karakter POKE kódja,

f a területen megjelenitendő karakter szinkódja (0 és 15 közötti érték).

Példa: Legyen a képernyő bal felső 20x20-as négyzete sárga A betükkel feltöltve.

Megoldás:

FILL 0,0,19,19,1,7

3.6. Egy terület átmozgatása

Egy már a képernyőn levő terület máshova másolását végzi el a MOVE utasitás.

Alakja:

MOVE a,b,c,d,e,f

ahol az első négy paraméter (a,b,c,d) a másolandó terület megadására szolgál (lásd FCHR, FCOL),

e az uj terület sorindexe,

f az uj terület oszlopindexe.

Hatása: Az a. sor b. oszlopában kezdődő c,d karakternyi területet átmásolja a képernyő e. sorának f. oszlopától kezdődően.

Példa: Mozgassuk át a képernyő bal felső sarkában levő 10x10-es négyzetes területet a jobb alsó sarokba.

Megoldás:

```
10 PRINT "SHIFT/CLR/HOME"
20 FOR I=1 TO 10
30 PRINT"1234567890"
40 NEXT I
50 MOVE 1,1,10,10,15,30
```

3.7. Karakter inverz képének kiirása

Egy képernyőterületen a karakterek inverz képének megjelenitésére szolgál az INV utasitás.

Alakja:

```
INV a,b,c,d
```

ahol a négy paraméter a terület megadását szolgálja, amelyre az INV utasitás vonatkozik (lásd FCHR, FCOL).

Hatása: az a. sor b. oszlopától kezdődő c karakter szélességü és d karakter mélységü területen a karakterek inverz képe jelenik meg.

Példa: A képernyő bal felső sarkában egy 20x20-as négyzeten a karakterek inverz képe jelenjen meg.

Megoldás:

```
INV 0,0,19,19
```

A MOVE-nél elmondott példában a jobb alsó négyzet karaktereinek inverz képét a következő utasitás jeleniti meg:

```
60 INV 15,30,10,10
```

3.8. Képernyő egy területének rollozása

A képernyő kitüntetett sorainak illetve oszlopainak folyamatos "tekerését" végzi a következő utasitás. A mozgatás történhet jobbra, balra, fel, le.

Alakja:

a) LEFT i a,b,c,d

b) RIGHTi a,b,c,d

c) UPi a,b,c,d

d) DOWNi a,b,c,d

ahol az első szó az utasitás irányát jelölő kulcsszó,

i a tekerés tipusa, értékei W vagy B,

a,b az oszlop és a sor sorszámai, ahonnan a mozgatás kezdődik,

c,d a mozgatandó terület utolsó oszlopának illetve sorának sorszámai.

Az i=W esetben a sorok (oszlopok) mozgatása ugy történik, hogy a karakterek nem tünnek el,

i=B esetben a sorokat nem "viszi" magával, azaz a mozgatott terület helyén üres hely lesz.

Az oszlopok és sorok számozása itt is nullával kezdődik.

Hatása:

a) Jobbról-balra "rolloz" az a. oszlop b. sorától kezdve a c. oszlop b. soráig bezárólag. A sorok illetve oszlopok tartalma az i paramétertől függ.

b) Hatása ugyanaz, de a mozgatás balról jobbra történik.

c) Hatása ugyanaz, de a mozgatás alulról felfelé történik.

d) Hatása ugyanaz, de a mozgatás felülről lefelé történik.

Megjegyzés: az utasitásokat nem használhatjuk finom grafikában!

Példa: a képernyő bal felső sarkán levő számok különböző irányu rollozását mutatjuk be az alábbi programmal.

```
 10  PRINT "SHIFT/CLR/HOME"
 10  FOR I=1 TO 10
 20  PRINT I,I*I,I*I
 30  NEXT I
 40  FOR I=1 TO 100
 50  LEFTW 0,0,10,10
 60  NEXT I
 70  FOR I=1 TO 100
 80  RIGHTW 0,0,10,10
 90  NEXT I
100  FOR I=1 TO 100
110  UPW 0,0,10,10
120  NEXT I
130  FOR I=1 TO 100
140  DOWNW 0,0,10,10
150  NEXT I
```

3.9. Képernyőtartalom kiirása

3.9.1. Képernyőtartalom kiirása háttértárolóra

Egy képernyő tartalmának háttértárolón való elhelyezésére szolgál az SCRSV utasitás.

Alakja:

a) SCRSV 2,8,2,"n,S,W"
b) SCRSV 1,1,1,"n,S,W"

ahol n a megőrzendő képernyő adatállomány neve a külső tárolón.

Hatás: az a) esetben az aktuális képernyőtartalmat mágneslemezre, a b) esetben mágnesszalagra teszi ki n névvel.

A továbbiakban ezzel a névvel lehet hivatkozni és ujra megjeleni - teni az adatokat (például az SCRLD utasitással).
Megjegyzendő, hogy az utasitás nem használható finom-grafikus mód esetén.

3.9.2. Képernyőtartalom beolvasása háttértárolóról

A külső SCRSV utasitással kiirt képernyőtartalom visszahozását végzi a SCRLD utasitás.

Alakja:

a) SCRLD 2,8,2,"n"
b) SCRLD 1,1,1,"n"

ahol az a) és b) alakban irt számok megegyeznek az SCRSV utasitásnál elmondottakkal,
n a tárolt képernyőtartalom azonositója.

Hatása: az n néven SCRSV utasitással kiirt képernyőtartalmat a) le- mezről, b) szalagról visszahozza és megjeleniti.

Példa: az aktuális képernyőtartalmat irassuk ki lemezre ALMA néven, majd olvassuk vissza és jelentessük meg a képernyőn.

A feladatot az alábbi program végzi el:

```
10  SCRSV 2,8,2,"ALMA,S,W"
20  PRINT "SHIFT/CLR/MOME"
30  SCRLD  2,8,2,"ALMA"
```

3.9.3. Képernyőtartalom kiirása nyomtatón

A képernyő tartalmát kiirathatjuk sornyomtatóra a COPY utasitással.
Alakja:

COPY

Hatása: az aktuális képernyőtartalmat a gép kiirja a nyomtatóra.

Megjegyzendők:

- az utasitást csak grafikus üzemmódban használhatjuk,
- a CIRCLE utasitással kirajzolt kör, ellipszis a kiirásnál torzulhat, ezért figyelembe kell venni az ott elmondottakat.

Nem grafikus képernyőtartalom nyomtatón való kiirását végzi a HRDCOPY utasitás.
Alakja:

HRDCOPY

3.10. Kurzor helyének meghatározása

A kurzor képernyőn levő helyének értékét adja meg a LIN függvény.
Alakja:

LIN

Értéke: numerikus, annak a sornak a száma, amelyikben a kurzor jelenleg van.

Megjegyzendő, hogy a függvény értékét programból is lekérdezhetjük, és direkt módban is használható.

4. Programírást segítő parancsok

4.1. Automatikus sorszámozás

Az utasitások számának automatikus generálását, kiirását és növelését végzi el az AUTO parancs.

Alakja:

AUTO s,n

ahol s a kezdeti sorszám,
n a növekmény.

Tekintsük a következő programot:

```
1000 INPUT A,B
1100 X=A+B
1200 PRINT X
```

A fenti programot a következő lépésekben készithetjük el az automatikus számozás felhasználásával:

Begépelés	Képernyő tartalma
AUTO 1000,100	1000
INPUT A,B	1000 INPUT A,B
RETURN	1000 INPUT A,B
	1100
X=A+B	....
	1100 X=A+B
RETURN	....
	1200
PRINT X	....
	1200 PRINT X
RETURN	....
	1300
RETURN	

4.2. Program átszámozása

A program átszámozását a RENUMBER parancs végzi.
Alakja:

RENUMBER s,n

ahol s az uj kezdőszám,
n a növekmény.

Hatására: a gép átszámozza a programot ugy, hogy az első sor az s számot kapja, minden további számozott sor az előzőtől n-nel lesz nagyobb.

Megjegyezzük, hogy a programban szereplő GOSUB, GO TO utasítások utáni számokat a gép automatikusan nem cseréli. Az átszámozás miatt bekövetkező esetleges programhibákat a SIMONS BASIC-ban levő PROC utasitás használatával küszöbölhetjük ki.

Tekintsük a következő programrészletet

```
10  INPUT X
20  Y=X+X
30  GO TO 100
```

Számozzuk át a fenti programrészletet ugy, hogy az első utasitás száma 1000 legyen, a továbbiak pedig 100-zal növekedjenek.
Ezt a következő paranccsal végezzük el

RENUMBER 1000,100

A LIST parancs végrehajtása után a következő programot kapjuk:

```
1000  INPUT X
1100  Y=X+X
1200  GO TO 100
```

4.3. Programok összemásolása

Programok összemásolására szolgál a MERGE parancs. A gép egy a gépben lévő programhoz tud ezzel hozzámásolni egy valamilyen tárolón (lemez, szalag) elhelyezett programot.

Alakja:

MERGE "n", d

ahol n a mágneses tárolón elhelyezett program neve,
d az egységszám (1 esetén kazettás magnetofon, 8,...,15 esetén lemez).

Hatása: a gépben aktuálisan helyetfoglaló programhoz hozzámásolja az n nevű programot, amit korábban a d számu egységre irtunk ki.

Vigyázzunk az összemásolandó programok sorszámaira, ugyanis a másolandó programot egyszerűen a gépben lévő program mögé helyezi el.

4.4. Programlistázás vezérlése

4.4.1. Listázás oldalanként

Egy program listázása a LIST paranccsal történik ugy, hogy a gép folyamatosan irja ki a programokat. A program megadott soronkénti egységekben történő listázását végzi a PAGE parancs.

Alakja:

PAGE n

ahol n az egyszerre képernyőre irandó sorok száma.

Hatása:

- a programot az első sorától kezdve n soronként kiirja a LIST parancs hatására,

- a következő n sor kiirását a RETURN billentyü lenyomásával végezhetjük el.

A parancsot a PAGE 0 parancs kiadásával hatástalanitjuk.
Példa: Irassuk ki a gépben levő programot 10 soronként.
Ezt az alábbi parancsokkal érhetjük el:

PAGE 10/RETURN
LIST/RETURN

A fenti két parancs hatására a gép kiirja a program első 10 sorát, a további sorokat a RETURN billentyü megnyomásával listázhatjuk ki tizes egységekben.

4.4.2. Listázás lassitása

A listázás lassitását a CTRL billentyü lenyomva tartásával érhetjük el. A jelen verzió egy további lassitási lehetőséget nyujt a DELAY paranccsal.
Alakja:

DELAY n

ahol n 1 és 255 közötti numerikus érték.

Hatása:

- a SHIFT/LOCK billentyüt lenyomva a DELAY parancs hatására a program listázása lelassul,
- leggyorsabb a listázás n=1 esetén,
- n=255 esetén a listázás karakterenként történik (alapsza- [illegible]

A billentyűt elengedve a listázás normál módban történik.
A COMMODORE billentyűt lenyomva a listázás megszakad, elengedve tovább folytatódik.

4.5. SIMON'S utasitások kiemelése

A SIMON'S BASIC egy sor "uj" utasitást tartalmaz a BASIC-hez képest. Az ezen utasitások megkülönböztetése lehetséges az eredeti utasitásoktól az OPTION paranccsal.
Alakja:

OPTION n

ahol n 0 és 255 közötti érték lehet.

Hatása: n=10 esetén a SIMON'S BASIC utasitások forditott (REVERSE) módban kerülnek kiiratásra listázásnál

n más megengedett értéke esetén a forditott mód kiiratás megszűnik.

4.6. Karakterlánc megkeresése

Egy BASIC programban előforduló tetszésszerinti karakterlánc helyének (helyeinek) megkeresésére szolgál a FIND parancs. Sok felhasználási területe közül csak egyet említünk meg: átsorszámozás esetén segitséget nyujt a GO TO, GOSUB utasitások megkeresésére és átsorszámozására!
Alakja:

FIND c

ahol c a keresendő karakterlánc.

A c értékei a következők lehetnek:

a) karakter konstans idézőjelbe téve
b) numerikus számjegy (számjegyek)
c) BASIC alapszó
d) változónév
e) egyéb jel/ek/.

Hatása a fenti esetekben: kiirja azon sorok sorszámát, ahol

a) a megadott karakterkonstans előfordul,
b) az adott számjeggyel (számjegyekkel) kezdődő konstans előfordul,
c) a megadott BASIC kulcsszó előfordul,
d) a megadott betüvel (vagy jelekkel) kezdődő változó előfordul,
e) egyéb jel/ek/ található/k/.

Példa: tekintsük a következő programrészletet:

```
10 INPUT  A1:X=15
20 INPUT  AØ
30 PRINT A1, AØ
40 A=A1*A1*1.5
50 PRINT "EREDMENY" , A1
60 GO TO 100
```

A FIND parancs végrehajtása

Parancs	Lista
FIND "EREDMENY"	50
FIND1	10,30,40,60
FIND PRINT	30,50

FINDA	10,20,30,40,50
FINDA1	10,30,40,50
FIND=	10,40
FIND=1	10
FINDTO	
FIND,	30,50

4.7. Programfutás nyomonkövetése

Egy program futása során szükséges lehet az utasitások végrehajtási sorrendjét ellenőrizni. Ezt végzi a TRACE parancs.
Alakja:

TRACE n

ahol n az utasitás paramétere,
n=0 esetben a TRACE befejezi a működését.
n=10, esetben parancs aktiv.

Hatása: A képemyő jobb felső sarkában egy területet lefoglal, ahova az éppen végrehajtott utasitás sorszámát irja be. Ha itt kiirási terület van, akkor ezt felülirja.

Ha a program befejezésekor néhány befejező utasitás sorszámát ki akarjuk iratni, akkor a RETRACE utasitást használjuk. A program befejezésekor ugyanis a képernyőről az utasitássorszámok eltünhetnek a TRACE utasitás alkalmazása esetén.

Alakja:

RETRACE

4.8. Változók tartalmának kiírása

A gépben levő program változói tartalmának a kiírására is lehetőség van a DUMP paranccsal.

Alakja:

DUMP

Hatása: A gép képernyőjén megjelennek a program változóinak nevei és értékei egyenlőségjellel összekötve. Amennyiben a változók száma 25-nél több, ugy a CTRL billentyűt lenyomva tartva kaphatjuk meg részenként a teljes listát.

Példa: Futtassuk le a következő programot:

```
10 A=25
20 B=A*2
30 C$="ALMA A FA ALATT"
```

Adjuk ki a DUMP parancsot.

A képernyőre a következők kerülnek kiírásra

```
A=25
B=50
C$="ALMA A FA ALATT"
```

5. Saját grafikus alakzatok készitése

5.1. Sprite manipuláció

Sprite, un. szellem /S/ létrehozása egyszerü utasitásokkal lehetséges

5.1.1. Helyfoglalás a memóriában

Az /S/ helyfoglalás - a MOB - részére a DESIGN utasitással lehetséges.

Alakja:

a) DESIGN a,b

b) DESIGN a,b+c

ahol

- a – a tipus jelölés, értékei 0 vagy 1,
- b – S részére lefoglalt tárolóhely első byte-jének cime,
- c – finom grafika esetén használatos konstans, értéke 49152 (v. $C000 hexadecimálisként).

Az a paraméter hatása

a=0 esetén finom grafika

a=1 esetén szines (multikolor) grafika

A b értékei:

- 64 többszörösének kell lennie,
- decimálisként vagy hexadecimálisként kell megadni, ez utóbbi esetben $ jellel kezdődik.

A kezdőcimek a következők lehetnek:

blokkszám	memóriahely
13-15	832-1023
32-63	2048-4o95
128-255	8192-16383

Ha a MEM utasitást is használjuk, akkor csak a 192-255 blokkok állnak rendelkezésre.

5.1.2. Szellem megrajzolása

Szellem megrajzolásához ad segitséget a @ eljárás.
Alakja:

a) @ ....

b) @ ....

A fenti pontok helyébe üthetjük be a szellem alakját kirajzoló karaktereket. Az a) formát a multikolor, a b)-t a finom grafika esetén alkalmazzuk. Mindkét esetben az egy alakzathoz tartozó sorok száma 21.
A beütendő karakterek értéke:

a) esetben, azaz multikolor módban

szinkód	értelmezése
B	a CMOB utasitás első szine
C	a MOB SET utasitásban megadott szin
D	a CMOB utasitásban megadott második szin

b) esetben

szinkód	értelmezése
B	a MOB SET utasitásban megadott szin

Természetesen mindkét esetben a kipontozva maradt terület a háttérszint veszi fel.

5.1.3. Szines sprite szineinek megadása

A multikolor módban a szinek megadását a CMOB utasitás végzi . Alakja:

CMOB s1, s2

ahol s1, s2 a megadott szinek kódja (0-15 közötti érték).

5.1.4. Szellem felépitése

A "szellem" paramétereit adhatjuk meg a MOB SET utasitással.

Alakja:

MOB SET a,b,c,d,e

ahol <u>a</u> a megadandó szellem száma, a kisebb számu nagyobb prioritásu, azaz eltakarja mozgatásnál az alacsonyabb prioritásut,

<u>b</u> a blokkszám, ahol a szellem adatai vannak,

<u>c</u> a szin megadása, amely szines szellem esetén a C, más esetben a B szinkódot jelöli ki,

d prioritás, amely hatására

d=0 érték esetén a szellem magasabb prioritású a képernyő egyéb jeleivel szemben,

d=1 esetben a helyzet forditott,

e tipus kijelölése

e=1 esetben szines,

e=0 esetben finom grafika.

5.1.5. Szellem megjelenitése és mozgatása

A szellem képernyőn történő megjelenitésére és mozgatására szolgál az MMOB utasitás.

Alakja:

MMOB a, x1, y1, x2, y2,b,c

ahol a a szellem száma,

x1,y1 a megjelenités helyének koordinátái,

x2,y2 azon pont koordinátái, ahova a mozgatás történik,

b a szellem méretének megadására szolgáló paraméter;

c mozgatási sebesség (1-255 közötti érték).

Hatása: a szellem képe az x1, y1 pontból az x2,y2 pontba mozog a b-ben megadott méretben c sebességgel.

Megjegyzések:

1. Az x1, y1, x2, y2 koordináták a szellem bal felső "sarkának" koordinátái.

2. A b méret az alábbi értékeket veheti fel:

érték	hatás
0	normál méret
1	x irányban kétszeres nagyitás
2	y irányban kétszeres nagyitás
3	mindkét irányban kétszeres méretnövekedés.

3. A mozgási sebesség ellentétes a máshol megadott sebességgel:

1	a leggyorsabb sebesség
255	a leglassabb sebesség .

A már a képernyőn levő szellem mozgatását végzi az RLOCMOB utasitás.

Alakja:

RLOCMOB a,x,y,b,c

ahol a megegyezik az MMOB utasitásnál elmondottakkal,
x,y azon hely koordinátái, ahova a szellemet mozgatjuk,
b,c paraméterek megegyeznek a MMOB utasitásnál elmondottakkal.

Hatása: A szellem a képernyőn lévő aktuális helyéről az x,y pontokba megy a b mérettel és a c sebességgel.

5.1.6. Alakzatok összeütközésének érzékelése

Az alakzatok összeütközésének lekérdezését késziti elő a DETECT utasitás.

Alakja:

DETECT n

ahol n az összeütközés tipusának megadására szolgáló paraméter.

Az n értékei:

n=0 esetben két szellem közötti összeütközés lekérdezésének előkészitését irjuk elő,

n=1 esetben egy szellem és a képernyőn levő egyéb karakterek összeütközése lekérdezésének előkészitését irjuk elő.

Megjegyzések:

1. A DETECT utasitást a CHECK utasitás előtt alkalmazzuk.
2. A DETECT utasitást kétszer kell kiadni:

- az első végrehajtása esetén az összeütközést jelző regiszter tartalma törlődik,
- a második végrehajtásánál a fenti regiszter az összeütközés regisztrálására készen áll.

A CHECK kérdezi le az összeütközés regiszter tartalmát.

Alakja:

a) IF CHECK (m1,m2) = 0 THEN u

b) IF CHECK (0) = 0 THEN u

ahol m1,m2 szellemek számai,

u végrehajtható utasitás.

Hatása:

a) Lekérdezi, hogy az m1 és m2 szellem összeütközése bekövetkezett-e.

b) Lekérdezi, hogy a szellem valamely képernyőn levő karakterrel összeütközött-e.

Ha összeütközés történt bármely (a) és (b) esetben, akkor az u utasitás hajtódik végre, egyébként az IF utasitást követő utasitás.

5.1.7. Szellem kitörlése

A képernyőn levő szellem kitörlését végzi a MOB OFF utasitás.
Alakja:

MOB OFF a

ahol a a képernyőről kitörlendő szellem száma.

5.2. Saját karakterek készitése

A saját karakterek készitéséhez először át kell másolni a karakter ROM-ot RAM-ba (MEM utasitás). Ezután meghatározzuk a DESIGN utasitással azt a karaktert, amit ki akarunk cserélni. Ezután szerkesztjük meg a @ utasitással a kivánt karaktert. Az alábbiakban ezeket az utasitásokat mutatjuk be.

5.2.1. Memória átmásolása

A memóriaterület ROM memóriájában elhelyezett karakterkészlet RAM memóriaterületre történő átmásolását végzi a MEM utasitás.
Alakja:

Hatása: A C64 karakterkészlete RAM területre másolódik.

Megjegyzések:

1. A RAM területen ezután tetszésszerinti saját karaktert tehetünk a régiek helyébe.

2. Áttolja a képernyő RAM-ot a $CC00 cimtől kezdve (a kernál mögé).

3. Bekorlátozza a MOB területet a 192-től a 255-ös blokkokra.

5.2.2. Kicserélendő karakter megadása

A kicserélendő karaktert adjuk meg a DESIGN utasitás segitségével.

Alakja:

DESIGN 2, $E000+C*8

ahol C a kicserélendő karakter poke kódja.

Hatása: a poke kóddal megadott karaktert kicserél a @ -ban megadott "formátumra", amit alább ismertetünk.

A @ segitségével szerkeszthetjük meg az uj karaktert.

Alakja:

@ ....

Megjegyzések:

1. Összesen 8 sort kell megadni egy karakter megszerkesztéséhez.

2. A helyére B betüt kell irni oda, ahol a karaktert meg akarjuk szerkeszteni.

Példa: egy karakter megszerkesztése:

```
50   @ . B . B B . B .
60   @ . B . B B . B .
70   @ . B . . . . B .
80   @ . B . . . . B .
90   @ . B . . . . B .
100  @ . B . . . . B .
110  @ . . B . . B . .
120  @ . . . B B . . .
```

A fenti karaktert a @ karakter helyett alkalmazzuk. Ehhez a következő utasításokat kell végrehajtani:

```
10   MEM
20   DESIGN 2,$E000+0*8
```

5.3. Példa MOB szerkesztésére, mozgatására, szinezésére

Készitsük el és futtassuk le az alábbi programot:

```
10   DESIGN 1,11*64
20   @ ............
30   @ ............
40   @ ............
50   @ ............
60   @ BBBBBBBBBBB.
70   @ .....B......
80   @ .....B......
90   @ ....BBB.....
100  @ ..DDDCC.....
110  @ ..D..C C....
120  @ .D...C.C....
130  @ .D...C.C....
140  @ D....C.C....
150  @ D....C.C...B
160  @ BBBBBBBB..BB
170  @ BBBBBBBBBBBB
180  @ BBBBBBBBB...
190  @ ..B..B......
200  @ .B....B.....
210  @ ............
220  @ ............
230  MOB SET 1,11,5,1,1: PRINT: SHIFT/CLR/HOME"
240  FOR T=1 TO 100
250  RLOCMOB 0,RND(1)*300+20,RND(1)*180+20,RND(1)*4,
     RND(1)*80+1
260  CMOB RND(1)*15,RND(1)*15:NEXT
```

6. Zenei utasitások

A C64 BASIC-ben az egyes zenei paramétereket a POKE utasitás felhasználásával vezérelhetjük. A SIMON'S BASIC lehetővé teszi a paraméterek utasitásban való megadását is.

6.1. Hangerő szabályozása

A hangerősség megadását teszi lehetővé a VOL utasitás.
Alakja:

VOL n

ahol n a hangerősség megadására szolgáló paraméter.
Az n értéke 0–15 között változhat a következőképpen:

n=0 esetén "lekapcsoltuk" a hangot,
n=1 minimális hangerő,
. . . .
n=15 maximális hangerő.

6.2. Hullámforma és egyéb paraméterek

Különböző paraméterek megadását érhetjük el a WAVE utasitással.
Alakja:

WAVE h, b

ahol h a hang megadására szolgáló paraméter,
b egy 8 jegyü kettes számrendszerbeli szám.

A h értékei:

1	1. hangra vonatkozik
2	2. hangra vonatkozik
3	3. hangra vonatkozik

A b egyes helyértékei egy-egy hang jellemző megadására szolgálnak. Ha az aktuális helyen 1 van, akkor a megfelelő effektus aktiv, egyébként nem.
Az egyes helyértékek a következőket jelentik (balról jobbra):

1. hely	zajgenerátor
2. hely	négyszög hullámforma
3. hely	fűrészfog hullámforma
4. hely	háromszög hullámforma
5. hely	nem használatos
6. hely	háromszög hullámforma esetén ring moduláció
7. hely	szinkronizáció
8. hely "kapu" bit,	a SIMON BASIC automatikusan beállitja

6.3. Burkoló görbe megadása

A burkoló görbe paramétereinek megadását végzi az 1-3 hang esetén az ENVELOPE utasitás.
Alakja:

ENVELOPE h, A, D, S, R

ahol h a hang paraméter (1,2 vagy 3)
A felfutás maximuma (Attach)
D lefutás az egyenletes erősségre (Decay)
S egyenletes erősség ideje (Sustain)
R lefutás nullára (Release).

6.4. Lejátszandó zene megszerkesztése

A zene tempójának és hangjainak megadására szolgál a MUSIC utasitás.
Alakja:

MUSIC n,"c"

ahol n az un. időtartam paraméter, a tempó megadására szolgál (1 és 255 közötti érték),
c lejátszandó zene megadására szolgáló karakterlánc.

A c tartalmazza a lejátszandó zenei hangokat. Első karaktere azt mondja meg, hogy melyik hangot szólaltatja meg (azaz az egyest, kettest vagy hármast). Ezután kell megadni a lejátszandó zenét. Minden egyes hangnál meg kell adni a

- hang hosszát egy vezérlő karakterrel,
- a hang nevét (C–H),
- az oktávot (0–7).

A vezérlő karakter értékeit és jelentését az alábbiakban foglaljuk össze:

vezérlő karakter	jelentése
SHIFT/CLR/HOME+1v.2v.3	a hang megadására szolgáló vezérlőjel
F1	következő hang 1/16 ideig szól
F3	következő hang 1/8 ideig szól
F5	következő hang 1/4 ideig szól
F7	következő hang 1/2 ideig szól
F2	következő hang 1/1 ideig szól
F4	következő hang 2/1 ideig szól
F6	következő hang 4/1 ideig szól
F8	következő hang 8/1 ideig szól

A hang és az oktáv megadásához a nevét az oktáv számát kell megadni.

Például: C5, D5, F2, E3

6.5. Zene lejátszása

A MUSIC utasitásban megadott zene lejátszására szolgál a PLAY utasitás.

Alakja:

PLAY n

ahol n értéke 0,1 vagy 2 lehet és a zene lejátszásának "módját" adja meg.

Az n értékeire és jelentésére tekintsük az alábbi táblázatot:

n értéke	jelentése
0	zene leállitása
1	megadott zene lejátszása a program futásának felfüggesztésével
2	megadott zene lejátszása a program párhuzamos futtatásával

A zenei lehetőségek bemutatására tekintsük az alábbi egyszerű mintaprogramot:

```
150 VOL 10
170 WAVE 1,00100000
190 ENVELOPE 1,0,15,10,0
210 A$="]1Z█C5█E5█F5█"
220 A2$="G5█C5█E5█F5█G5█C5█E5█F5█G5█"
230 A2$=A2$+"E5█C5█E5█D5█E5█E5█D5█C5█"
240 A2$=A2$+"C5█E5█G5█G5█F5█F5█E5█F5█"
250 A2$=A2$+"G5█E5█C5█D5█C5█C5█C5█E5█"
260 A2$=A2$+"F5█"
270 A3$="C5█C5█"
280 MUSIC 8,A$+A2$+A2$+A3$
300 PLAY 1
320 VOL 0
330 END

READY.
```

7. Egyéb parancsok

7.1. Disk müveletek

A SIMON BASIC-ben a lemezmüveletekhez szükséges parancsok köre is bővül. A disk müveletek egy része helyettesithető a DISK parancs-csal.

Alakja:

DISK "müvelet"

ahol a müvelet logikai állományok megnyitása, egységszám, második cim megadása, PRINT logikai állomány, program el-tevése stb. lehet.

Példák:

1. Lemez header készités TESZT néven

DISK "NO:TESZT,01"

2. Lemezen levő ALMA nevü program törlése

DISK "SO:ALMA"

7.2. Lemezállományok neveinek listázása

A lemezen levő állományok neveinek listázását végezhetjük a DIR pa-ranccsal.

Alakja:

a) DIR "Ø

b) DIR "Ø:s*

c) DIR "Ø:ks*

ahol $ dollár jel,

k kérdőjel vagy kérdőjel sorozat

s karakterlánc.

Hatása:

a) Kiirja a gépben levő lemez direktoriját (lásd LOAD "$",8).
b) Kiirja az s karakterlánccal kezdődő állományokat.
c) Kiirja azon állományokat, amelyek nevében a kérdőjelek számát meghaladó karakterek megegyeznek s-sel.

Példa: Irassuk ki azon állományok neveit és helyfoglalását, amelyek nevének 3. és 4. karaktere PQ

Megoldás:

DIR "$:??PQ*

A DIR paranccsal számos egyéb "feladat" is megoldható. A 4 karakter, illetve a 9 karakter hosszuságu programok nevét a következő utasitásokkal irhatjuk ki:

DIR "$:????
DIR "$:?????????

7.3. NEW parancs hatástalanitása

Amennyiben kiadtuk a NEW parancsot, de valamilyen okból mégis szükségünk van az utoljára a gépben levő programra, akkor az OLD paranccsal hozzáférhetünk.
Alakja:

OLD

azaz paramétere a NEW parancshoz hasonlóan nincs. Egy alkalmazást megemlitünk: amennyiben a COLD parancsot használjuk, ez a gépben lévő aktuális programot is "kitörli". Az OLD paranccsal azonban ez is aktivizálható.

7.4. Funkció billentyük használata

Funkció billentyük parancshoz rendelése.

Egy parancs funkció billentyühöz való rendelését végzi a KEY utasitás.

Alakja:

KEY s,"c"

ahol s a funkció billentyü száma ($1 \leq s \leq 16$)

c utasitásnév.

Az utasitásnév legfeljebb 16 karakter hosszu lehet. A KEY parancsot a kulcsszó legépelésével és a RETURN billentyü lenyomásával lehet végrehajtani. A továbbiakban a megfelelő funkció billentyüt lenyomva a hozzátartozó c szöveg jelenik meg a képernyőn.

Példa:

KEY 1, "LOAD"

Az utasitás végrehajtása után az F1 billentyüt lenyomva a LOAD kiirás jelenik meg a képernyőn.

A kiirt parancs végrehajtása csak akkor történik meg, ha a RETURN billentyüt is lenyomjuk. Ezen billentyü "lenyomását" is hozzárendelhetjük a KEY utasitással valamelyik funkcióbillentyühöz ugy, hogy

+CHR$(13)

Példa:

KEY 1,"LOAD"+CHR$ (13)

A fenti parancs végrehajtása után az F1 billentyü megnyomása ekvivalens lesz a következő billentyük megnyomásával:

L, O, A, D, RETURN

A képernyőre kiirathatjuk az egyes funkció billentyükhöz rendelt utasitás - neveket is a DISPLAY paranccsal.

Alakja:

DISPLAY

A parancs végrehajtása a RETURN billentyü lenyomásával történik.

Tételezzük fel, hogy a fenti KEY parancsot már végrehajtottuk, és a DISPLAY parancsot begépeltük. A gép képernyőjén a következő szöveg jelenik meg:

```
DISPLAY:KEY 1,"LOAD"+CHR$(13)
KEY2,""
KEY3,""
KEY4,""
KEY5,""
KEY6,""
KEY7,""
KEY8,""
KEY9,""
KEY10,""
KEY11,""
KEY12,""
KEY13,""
KEY14,""
KEY15,""
KEY16,""
```

Billentyűk használata:

F1, F3, F5, F7	billentyű megnyomása,
F2, F4, F6, F8	F1,F3,F5,F7 billentyű és SHIFT
F9, F11, F13, F15	F1,F3,F5,F7 billentyű és a COMMODORE billentyű
F10, F12, F14, F16	F1,F3,F5,F7 billentyű és SHIFT és COMMODORE billentyű

III.

SUPERGRAPHIK 64

Bevezetés

A rajzolási lehetőségek kényelmesebbé tételére készült a

SUPERGRAPHIK 64

software anyag. A gyártó a DATA BECKER cég.

Két változat létezik:

a) mágnesszalagos
b) mágneslemezes

A mágnesszalagos változat valamivel kevesebb lehetőséget tartalmaz, mint a mágneslemezes változat. A fő különbség az, hogy a b) változat zenei utasításokkal és az un. 80x50-es grafikával bővült.

Mivel hazánkban a szalagos változat terjedt el elsősorban, ezért ezt ismertetjük részletesen. A lemezes változat két eltérő utasításcsoportját (zene, 80x50-es grafika) részlete - sen ismertetjük, a többi utasítást azonban csak vázlatosan.

1. Szalagos változat

1.1. Rendszer betöltése

Betöltéséhez az alábbi müveletek szükségesek:

1. Helyezzük a SUPERGRAFIK64 kazettát a kazettás egységbe
2. Gépeljük be a

   LOAD

   parancsot.

   A gép válasza:

   PRESS PLAY ON TAPE

3. Néhány fordulat (~7) után a képernyőn megjelenik a

   FOUND GRAFIK 64

   üzenet.

4. Majd a képernyő megint eltünik és a program betöltése után kiirja a

```
CBM64 HI-RES GRAPHICS SOFTWARE REV.F
/C/ BY DATA BECKER GMBF
CREATED 19/01/1983
BY M. THIELKER
GRAPHICS READY
READY
```

üzenetet.

1.2. Rajzolási mód megadása

Rajzolási mód megadása nélkül nem jelenik meg rajz a képernyőn. Ezért a rajzolás megkezdése előtt ki kell adni a mód megadására szolgáló GRAPHICS parancsot.

Alakja:

!GRAPHICS p

ahol p a rajzolási paraméter.

A p a következő értékeket veheti fel:

p	jelentése
0	a grafikus képernyőről visszaállunk normál módra
1	HR grafika kapcsolódik be, a további utasitások erre vonatkoznak
2	megegyezik a p=1-gyel
3	a HR grafika bekapcsolódik, a következő utasitások azonban a normál képernyőre vonatkoznak
4	megegyezik a p=3 esettel
5	a HR grafikus képernyő kikapcsolódik, de az utasitások erre vonatkoznak, a grafikus kép akkor jelenik meg ujra, ha kiadjuk a GRAPHICS 1 parancsot

A fenti táblázatban a HR (high resolution graphics) a grafikus mó- dot jelenti.
Az utasitás részletes értelmezését a következő példákban mutatjuk be.

1.3. Grafikus képernyő törlése

Az aktuális grafikus képernyő tartalmának kitörlésére szolgál a GCLEAR utasitás.
Alakja:

!GCLEAR

1.4. Geometriai alakzatok megjelenitése

1.4.1. Egy pont megjelenitése, törlése

Egy pont megjelenitésére szolgál a DOT utasitás.
Alakja:

!DOT x,y

ahol x,y a megjelenitendő pont koordinátái.

Egy pont törlésére szolgál a CDOT utasitás.
Alakja:

!CDOT x,y

ahol x,y a kitörlendő pont koordinátái.

Példa: Tekintsük a következő programot:

```
10  !GCLEAR:REM KEPERNYO TORLES
20  !GRAPHICS 1: REM MOD MEGADAS
30  FOR T=1 TO 100
40  !DOT 2*T,T : REM 1.EGYENES RAJZOLAS
50  !DOT 2*T, T+5 : REM 2.EGY.RAJZ.
60  !DOT 2*T, T+10 : REM 3.EGY.RAJZ.
70  !DOT 2*T, T+15 : REM 4.EGY.RAJZ.
80  !DOT 2*T, T+20 : REM 5.EGY.RAJZ.
90  NEXT T
```

A fenti program 5 párhuzamos egyenest rajzol a képernyőre. A legutolsó vonal felső felét törli ki a következő programrészlet:

```
100  FOR T=51 TO 0 STEP-1
110  !CDOT 2*T, T+20 : REM TORLES
120  NEXT T
```

1.4.2. Egyenes szakasz kirajzolása és törlése

Egyenes kirajzolását érhetjük el a LINE utasitással.
Alakja:

!LINE x1,y1 TO x2,y2

ahol x1,y1 a kirajzolandó szakasz kezdőpontjának koordinátái,
x2,y2 a kirajzolandó szakasz végpontjának koordinátái.

Példa: Tekintsük a következő programot

```
10   !GCLEAR
20   !GRAPHICS 1
30   FOR J=0 TO 199 STEP 5
40   !LINE 0,J TO 319,J:REM EGYENES RAJZ.
50   NEXT J
```

A fenti program vizszintes vonalakkal rajzolja tele a képernyőt.
Ha függőleges vonalakkal is tele akarjuk "irni" a képernyőt, akkor a következő programot kell még lefuttatnunk:

```
60   FOR I=0 TO 319 STEP 5
70   !LINE I,0 TO I,199
80   NEXT I
```

Az egyenes szakasz törlését végzi el a CLINE utasitás.
Alakja:

!CLINE x1,y1 TO x2,y2

ahol x1,y1 a törlendő egyenes kezdőpontjának koordinátái,
x2,y2 a törlendő egyenes végpontjának koordinátái.

Példa:
A fenti programmal kirajzolt vizszintes és függőleges egyenesek minden másodikát törölhetjük ki, ha a következő programot futtatjuk le:

```
90   FOR J=0 TO 199 STEP 10
100  !CLINE 0,J TO 319,J
110  NEXT J
120  FOR I=0 TO 319 STEP 10
130  !CLINE I,0 TO I,199
140  NEXT I
```

Szaggatott vonallal rajzolhatjuk ki a szakaszt a DLINE utasitással.

Alakja:

!DLINE x1,y1 TO x2,y2

ahol x1,y1 a szakasz kezdőpontjának koordinátái,
x2,y2 a szakasz végpontjának koordinátái.

A grafikus kurzor aktuális pontjától rajzol egy szakaszt a megadott pontig a DRAW utasitás.

Alakja:

!DRAW x,y

ahol x,y a megrajzolandó egyenes szakasz végpontja.

Megjegyezzük, hogy a grafikus kurzort rajzolás nélkül is tudjuk mozgatni. Erre szolgál a MOVE utasitás.

Alakja:

!MOVE x,y

ahol x,y azon pont koordinátái, ahova a grafikus kurzor mozgatása történik (természetesen rajzolás nélkül!).

1.4.3. Téglalap kirajzolása, eltörlése

Egy téglalapot rajzolhatunk ki a FILL utasitás segitségével, amely egyben be is festi az adott rajzolási szinnel a téglalapot.

Alakja:

!FILL x1,y1 TO x2,y2

ahol x1,y1 a téglalap bal felső sarkának koordinátái,
x2,y2 a téglalap jobb alsó sarkának koordinátái.

Megjegyzések:

1. A téglalap oldalai párhuzamosak a rajzolási területtel (képernyő oldallapjai).
2. A kitöltési szin alapállapotban fehér.
3. Szines téglalapokat a LCOL és a BCOL utasitások előzetes használatával rajzolhatunk (lásd későbbi pontban).

Ha nem akarjuk a téglalapot befesteni, csak az oldalait kirajzolni, akkor a FRAME utasitást használjuk.

Alakja:

!FRAME x1,y1 TO x2,y2

ahol x1,y1 a kirajzolandó téglalap bal felső sarkának koordinátái,
x2,y2 az ezen téglalap jobb alsó sarkának koordinátái.

A FILL utasitással kirajzolt "befestett" téglalap eltüntetésére szolgál a CLEAR utasitás.

Alakja:

!CLEAR x1,y1 TO x2,y2

ahol x1,y1 a kitörlendő téglalap bal felső sarkának koordinátái,
x2,y2 az ugyanezen téglalap jobb alsó sarkának koordinátái.

A FRAME utasitással kirajzolt téglalap kitörlését végzi a CFRAME utasitás.

Alakja:

!CFRAME x1,y1 TO x2,y2

ahol az utasitás paraméterei megegyeznek a CLEAR utasitás paramétereivel.

Példa:

A fenti utasitásokat felhasználva rajzoljunk ki egy sakktáblát.

A feladatot a következő programmal oldjuk meg:

```
 10  !GCLEAR
 20  !GRAPHICS 1
 30  !FRAME 0,20 TO 160,180
 35  FOR I=1 TO 8 STEP 2
 40  FOR T=0 TO 140 STEP 40
 50  !FILL T,I*20 TO T+20,I*20+20
 60  NEXT T
 70  FOR T=20 TO 160 STEP 40
 80  !FILL T,I*20+20 TO T+20,I*20+40
 90  NEXT T
100  NEXT I
```

1.4.4. Szabályos görbevonalu alakzatok rajzolása

Kör, ellipszis rajzolására szolgál a CIRCLE utasitás. Ezzel kirajzolhatjuk az alakzat egy részét is.
Alakja:

!CIRCLE x1,y1,x2,y2 /s1,s2/

ahol x1,y1 a kirajzolandó alakzat középpontja,
x2 az X irányu sugár hossza,
y2 az y irányu sugár hossza,
s1 a rajzolás kezdőszöge fokokban,
s2 a rajzolás "vég"-szöge fokokban.

Példa:

A képernyő közepére rajzol ki 5 koncentrikus kört a 159, 99 koordinátákkal megadott középponttal az alábbi program:

```
10   !GCLEAR
20   !GRAPHICS 1
30   FOR T=0 TO 100 STEP 20
40   !CIRCLE 159, 99, T,T
50   NEXT T
```

A szögek használatára módositsuk a programot az alábbi utasitásokkal:

```
30   FOR T=0 TO 100 STEP 5
40   !CIRCLE 159, 99, T, T, 100-T, 260+T
```

A lefuttatás után egy csepp alakot hagy ki a koncentrikus körök felső részéből a T-vel módositott szögeknek megfelelően.

A CIRCLE utasitással kirajzolt körök (ellipszisek) törlését végzi a CCIRCLE utasitás.

Alakja:

```
!CCIRCLE x1,y1,x2,y2 /,s1,s2/
```

ahol a megadott paraméterek megegyeznek a CIRCLE utasitásnál megadott paraméterekkel.

Példa:

Töröljük ki a fenti programmal megrajzolt körök minden másodikát.

Megoldás:

```
60   FOR T=0 TO 100 STEP 10
70   !CCIRCLE 159, 99, T, T, 100-T, 260+T
80   NEXT T
```

A CIRCLE utasitással kirajzolható alakzatokat szaggatott vonallal rajzolja ki a DCIRCLE utasitás.
Alakja:

```
!DCIRCLE x1,y1,x2,y2 /,s1,s2/
```

ahol az utasitás paraméterei megegyeznek a CIRCLE utasitás paramétereivel.
Példa:
Rajzoljuk ki szaggatott vonallal közös középpontu ellipsziseket.
Az alábbi program oldja meg a feladatot:

```
10  !GCLEAR
20  !GRAPHICS 1
30  FOR T=20 TO 100 STEP 20
40  !DCIRCLE 159, 99, T, T-20
50  NEXT T
```

A futtatás eredménye 5 ellipszis, amelynek a középpontja a 159, 99 pont, és a kirajzolás szaggatott vonallal történik.

1.5. Szinezési utasitások

1.5.1. Háttérszin megadása

A képernyő háttérszinének megadására szolgál a BCOL utasitás.
Alakja:

```
!BCOL n
```

ahol n a képernyő hátterének szine.

Emlékeztetőül a szinkódok:

0	fekete	8	narancs
1	fehér	9	barna
2	piros	10	világos piros
3	türkiz	11	1.sz.szürke
4	lila	12	2.sz.szürke
5	zöld	13	világoszöld
6	kék	14	világoskék
7	sárga	15	3.sz.szürke

1.5.2. Rajzolási szin megadása

Az alakzatok kirajzolásához megadhatjuk a rajzolási szint az LCOL utasitással.

Alakja:

!LCOL n

ahol n a rajzolási szin (0 - 15 közötti érték).

Hatása: az alakzatok egészen addig a megadott szinnel lesznek kirajzolva, ameddig ujabb LCOL utasitás nem következik.

Példa: Tekintsük az alábbi programot

```
10   !GCLEAR
20   !GRAPHICS 1
30   FOR T=0 TO 15
40   !BCOL T
50   !LCOL T
60   !FILL T*20,0 TO T*20+20,199
70   NEXT
```

Eredmény: a háttérszinek végigskálázzák az összes szint, miközben minden váltásnál egy ujabb függőleges szines téglalap rajzolódik az előzőek mellé.
A téglalapok végül kitöltik az egész képernyőt, a szinek sorrendje megegyezik a növekvő numerikus értékekhez tartozó szinekkel.

Megjegyzés: a keretszint továbbra is a POKE 53280,i utasitással adhatjuk meg!

Ha a keretszint is meg akarjuk változtatni a háttérszin szinére, akkor az 50-es utasitást cseréljük ki az alábbira:

```
50  !LCOL  T : POKE  53280,T
```

1.6. Szellem grafikai utasitások

A szellem kirajzolásához szükséges alakzat megadását szolgálja a SPRITE utasitás.

Alakja:

```
!SPRITE  n,v
```

ahol n a szellem száma (0–7 közötti érték),
v karakter tipusu változó, amely a szellem adatait tartalmazza (63 byte hosszuságu).

A szellem méretének és szinének megadását végzi az SMODE utasitás.

Alakja:

```
!SMODE  n,s,p
```

ahol n a szellem sorszáma
s a megjelenités nagysága

s = 0 normál nagyság,

s = 1 x irányban kétszeres nagyitás,

s = 2 y irányban kétszeres nagyitás,

s = 3 minden irányban kétszeres nagyitás,

p szellem szinének megadása (0-15 közötti egész szám).

Szellemgrafika mozgatására szolgál az SMOVE utasitás.
Alakja:

!SMOVE n,x,y

ahol n a szellem sorszáma,
x,y a mozgatás végpontjának koordinátái.

Példa: Az alábbi program a fenti utasitások használatát mutatja be. Az alakzat megszerkesztése után nagyitjuk x majd y irányba, ezután mindkét irányba. Bemutatjuk a szinek változtatásának módját és a mozgatás lehetőségét is.

```
1    !BCOL1
10   ADATØ=""
20   FORT=ØT062
30   READ ADAT
40   ADATØ=ADATØ+CHRØ(ADAT)
50   NEXT
60   !GCLEAR
70   !GRAPHICS1
80   !SPRITEØ,ADATØ
86   FORJ=ØT03
90   FORT=ØT0199
100  !SMODE Ø,J,J+2
110  !SMOVEØ,T,T
```

```
120   NEXT
130   NEXT
1000  DATA 255,Ø,255
1010  DATAØ,255,Ø
1020  DATA255,Ø,255
1030  DATAØ,255,Ø
1040  DATA255,Ø,255
1050  DATAØ,255,Ø
1060  DATA255,Ø,255
1070  DATAØ,255,Ø
1080  DATA255,Ø,255
1090  DATAØ,255,Ø
1100  DATA255,Ø,255
1110  DATAØ,255,Ø
1120  DATA255,Ø,255
1130  DATAØ,255,Ø
1140  DATA255,Ø,255
1150  DATAØ,255,Ø
1160  DATA255,Ø,255
1170  DATAØ,255,Ø
1180  DATA255,Ø,255
1190  DATAØ,255,Ø
1200  DATA255,Ø,255
1210  DATAØ,255,Ø
```

1.7. Egyéb utasitások

1.7.1. Képernyő tartalom tárolása és visszaolvasása

Az aktuális képernyőtartalmat háttértárolóra is kitehetjük a GSAV utasitással.

Alakja:

```
!GSAV"n",d
```

ahol n a kiirandó kép azonositója (egy karaktersorozat, mint a programnév!),

d a háttértároló egységszáma (1 esetben szalag, d=8-15 esetben lemez).

A képtartalom háttértárolóról történő visszahozására szolgál a RECALL utasitás.

Alakja:

```
!RECALL "n",d
```

ahol n és a d paraméterek megegyeznek a GSAV utasitásnál elmondottakkal.

Példa: A két utasitás illusztrálására tekintsük az alábbi szineket megadó programot.

```
10   !GCLEAR
20   !GRAPHICS 1
30   FOR T=0 TO 14
40   !LCOL T
50   !FILL T*20,0 TO T*20+20,199
60   NEXT
70   !GSAV "RAJZ-SZINEK", 8
80   GET A$:IF A$="" THEN 80
90   !GCLEAR:!GRAPHICS 1
100  !RECALL"RAJZ-SZINEK",8
```

A program a képernyőre kirajzolt képet elteszi mágneslemezre, majd egy tetszésszerinti karakter leütése után a képernyőt kitisztázza és visszahozza lemezről és megjeleniti.

Amennyiben a képet szalagra akarjuk kitenni, ugy a következő-ket kell tennünk:

1. Az egységszámot 8-ról 1-re kell cserélni a 70-es és 100 -as utasitásokban.

2. Be kell tenni egy kazettát a magnetofonba, és a rajzolás be-fejezése után be kell a PLAY és RECORD gombokat nyom - ni.

3. A felvétel befejezése után a képernyőtartalom ujra megjele - nik. Ekkor a szalagot vissza kell tekerni a felvétel elejére.

4. Meg kell nyomni egy tetszésszerinti billentyűt a gépen /GET...!/.

5. Le kell nyomni a PLAY billentyűt a magnetofonon.

6. A beolvasás után a szineket tartalmazó képernyő ujra megjele-nik.

Megjegyzés: a 8K grafikus képernyőtartalom kiirásához összesen 37 blokk szükséges, a SUPERGRAFIK tehát összesen 5 to-vábbi blokkot foglal a 8K képernyőtartalomhoz.

1.7.2. Képernyőtartalom kiirása nyomtatón

Grafikus képernyőtartalom kilistázását végzi a HARD utasitás. Alakja:

!HARD n

ahol n a printer egységszám (4 vagy 5).

Eredménye: A grafikus képernyőtartalmat kiirja a nyomtatóra.

Megjegyzés: Az utasitás használata előtt az OPEN utasi - tást ki kell adni a printerre.

Példa: Az aktuális képernyőtartalom kiirásához bárhova a programba tegyük be a következő utasitásokat:

```
OPEN 4,4
!HARD   4
```

Eredmény: az aktuális grafikus képernyőt kinyomtatja.

1.7.3. Egy képernyő pont ellenőrzése

Egy képernyő pont tesztelését végzi a TEST utasitás. A cél annak a kérdésnek az eldöntésére, hogy az aktuális képernyőn a vizsgált pontban történt-e rajzolás vagy sem.

Alakja:

```
!TEST x,y,v
```

ahol x,y a vizsgált pont koordinátái,

v a változónév, ahova a tesztelés eredményének értéke kerül:

v=0, ha nincs a pont kirajzolva,

v=1 egyébként.

1.7.4. Képernyőtartalmak cserélése

Lehetőség van grafikus képernyőtartalmak változtatására is ugy, hogy egy előbbit is ki tudunk rajzoltatni.

Alakja:

!TRANS

Mondanivalónk alaposabb megértéséhez futtassuk le és analizáljuk az alábbi programot.

```
 10   !BCOL 2
 20   !LCOL 0
 30   POKE 53280,2
 40   !GCLEAR
 50   !GRAPHICS 5
 60   !CIRCLE 100,100,20,50
 70   !TRANS
 80   !GCLEAR
 90   !CIRCLE 100,100,50,20
100   !GRAPHICS 1
110   FOR T=1 TO 50
120   !TRANS
130   NEXT T
```

1.7.5. Alakzatok kirajzolása

Egyenes részekből összeállitott tetszésszerinti alaku geometriai ábrák kirajzolását végzi a DISP utasitás.

Alakja:

!DISP p,x,y,s

ahol p rajzolási paraméter:

p=0 esetén az s alakzat kitörlődik,

p=1 esetén az s alakzat kirajzolódik,

x,y a kirajzolandó alakzat első elemének koordinátái,

s a kirajzolandó alakzatot megadó karakterlánc.

Példa:

A következő program egy háromszöget rajzol ki a képernyőre, majd egy billentyű lenyomása után eltünteti

```
10  !BCOL 2
20  !LCOL 0
30  A$="X]0]0]0]0]0]0]XQXQ]XQ]X
         Q]XQ]XQ]X||X||X||X||X||X||X||X||X||X||X||X"
40  !GCLEAR
50  !GRAPHICS 1
60  !DISP 1,100,100,A$
70  GET B$:IF B$=""GO TO 70
80  !DISP 0,100,100,A$
```

1.7.6. Szöveg kiiratása grafikus képernyőre

Szöveget a képernyőre kiirni a TEXT utasitással lehet.

Alakja:

```
!TEXT "s",x,y,p /,c/
```

ahol s a kiirandó szöveg,

x,y a kiiratandó szöveg kezdetének koordinátái,

p paraméter értékei:

p=0 esetben a szöveg kiiródik (nagybetű/grafika),

p=1 esetben a szöveg kiiródik (nagy/kisbetű),

c opcionális paraméter, a szöveg szine.

Példa:

Az 1.7.5. pontban közölt programot bővitsük a következő utasitásokkal:

```
55  !TEXT "EZ EGY HAROMSZOG",20,20,0
75  !TEXT"             VOLT!!!",20,20,0
```

1.7.7. Supergrafika kikapcsolása

A SUPERGRAPHIK 64 szalagos változatának kikapcsolását végzi az OFF utasitás.

Alakja:

```
!OFF
```

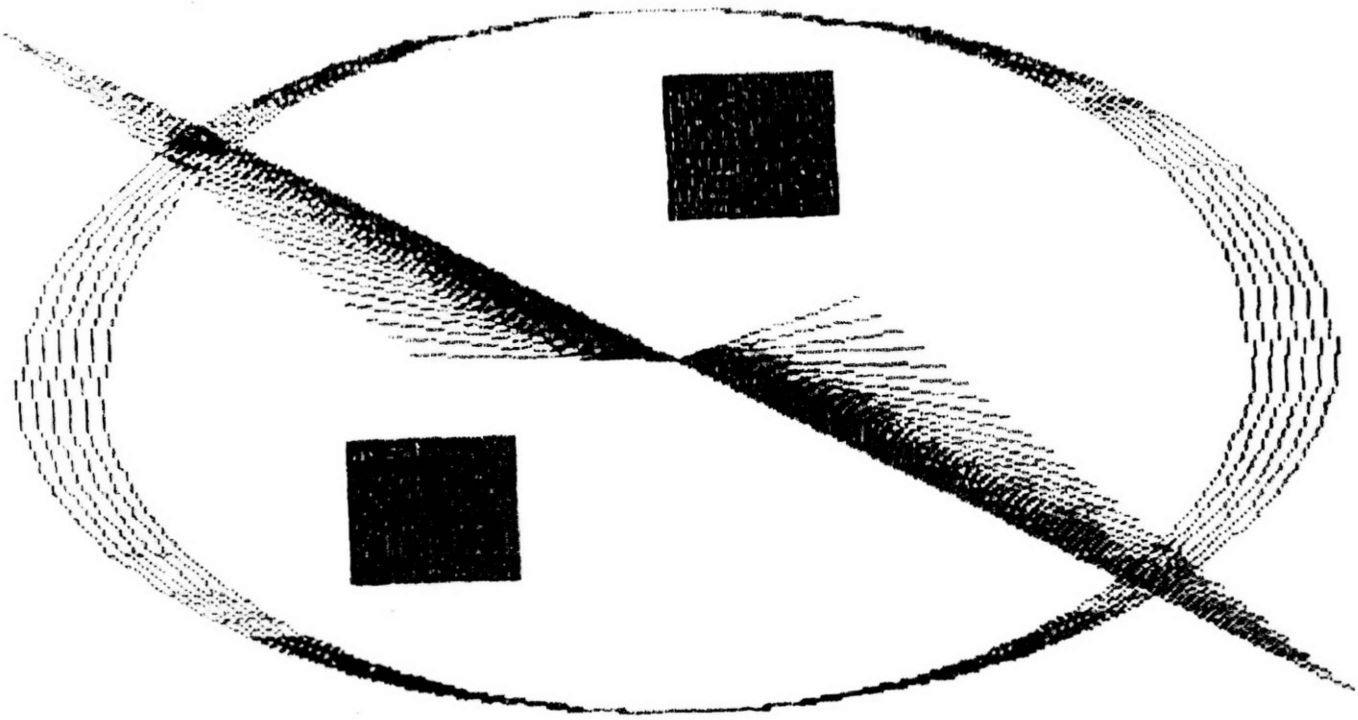
1.8. Példa program

Az alábbiakban bemutatunk egy programot, amely a rendszer utasításaiból többet használ, és érzékelteti a sokoldalu lehetőségeket .

```
10  POKE5328Ø,1
20  !BCOL 1
30  !LCOL 6
40  !GCLEAR
50  !GRAPHICS 1
60  FORT=16Ø T014Ø  STEP-4
70  !CIRCLE16Ø,1ØØ,T,1ØØ,9Ø,27Ø
80  !CIRCLE16Ø,1ØØ,T-2,1ØØ,27Ø,36Ø
90  !CIRCLE16Ø,1ØØ,T-2,1ØØ,Ø,9Ø
100 NEXT T
110 !LCOL 2
120 FORT=1T01ØØSTEP5
130 !MOVE16Ø,1ØØ
140 !DRAWT,T
150 NEXT
160 FORT=319T02ØØSTEP-5
170 !MOVE16Ø,1ØØ
180 !DRAWT,T-12Ø
190 NEXT T
200 FORT=ØT015
```

```
210  !LCOL T
220  !FILL8Ø,12ØTO12Ø,16Ø
230  !FILL16Ø,2ØTO2ØØ,6Ø
240  NEXT
250  !GCLEAR
260  !GRAPHICS 1
270  !BCOLØ:POKE5328Ø,Ø
280  !LCOL1
290  FORT=13ØTO18Ø
300  !LINET,2ØØ-T/2TO 2ØØ-T/3,T
310  !LINET,2ØØ-T/2TO 2ØØ-T/4,T
320  !LINET,2ØØ-T/3TO 2ØØ-T/4,T
330  !CLINET,2ØØ-T/2TO 2ØØ-T/3,T
340  !CLINET,2ØØ-T/2TO 2ØØ-T/4,T
350  !CLINET,2ØØ-T/3TO 2ØØ-T/4,T
360  NEXT
370  !GCLEAR
380  !GRAPHICS 1
390  POKE5328Ø,2:!BCOL2:!LCOL1
400  FORT=8ØTO2ØØ
410  !MOVE16Ø,1ØØ
420  !DRAWT/2,T/2+INT(J)
430  !CLINE16Ø,1ØØTOT/2,T/2+INT(J)
440  J=J+Ø.4
450  NEXT
460  !GRAPHICS Ø
470  END
```

A program egyik alakzatának rajzát alább közöljük:



2. Lemezes változat

A SUPERGRAPHIK 64 . legutóbb megjelent lemezes változata némileg eltér a korábbi, szalagos változattól.

Az utasitások között átfedés van. A lemezes változat elsősorban azzal több a szalagos változatnál, hogy

- a zenei lehetőségekkel is kibővitették,
- létezik egy 80x50-es grafikus képernyő változata is.

2.1. Rendszer aktivizálása

A SUPERGRAPHIK lemezes változatát a következőképpen töltjük be a gépbe:

1. Bekapcsoljuk a lemezegységet (és természetesen a gépet is).
2. Begépeljük a következőket:

   LOAD "*",8 /RETURN

   A gép üzenete néhány másodperces lemezforgás után / 3 mp/:

   READY

3. Begépeljük a

   RUN / RETURN

   parancsot.
   A floppy működésbe kezd, majd körülbelül 22-23 mp után a képernyő fekete lesz és rózsaszin betükkel a következő üzenetet irja ki:

   SUPERGRAPHIK CBM 64
   C 1983 BY DATA BECKER

Ezután használhatjuk a rendszer utasitásait. A továbbiakban összefoglaljuk a különböző utasitásokat. Részletes leirást a kézikönyvben találunk.

2.2. Zenei utasitások

A zenei utasitások köre nem tul széles és a következőkből áll:

Hangerő:

VOLUME = v

A SIMONS' BASIC VOL utasitásához hasonlóan a hangerő beállitására alkalmas. A v értéke 0 és 15 közötti érték lehet. Nullánál "kikapcsol", 15 a legerősebb hang.

2.2.1. Hangjellemzők megadása

A hangjellemzők megadását a SOUND utasitással érhetjük el. Alakja:

SOUND s,w,a,d,p,r,(f (p))

ahol s a hang megadása /1., 2 vagy 3. hangra vonatkozik az utasitás/,

w a hullámformát rögziti /háromszög, négyszög, fürészfog, zaj/,

a felfutás nulláról maximumra /attack/,

d lefutás átlagszintre /decay/,

p átlagszint ideje /sustain/,

r lefutás átlagszintről nullára (release),

f (p) szűrő paraméter.

2.2.2. Szürő beállitása

A szürő paramétereinek beállitására szolgál a FILTER utasitás.
Alakja:

FILTER a (,b,c)

ahol a a szürő jellemző megadására szolgáló paraméter (0-7 közötti érték),

b szürő határfrekvencia (0-2047 közötti érték),

c szürő rezonancia frekvencia (0-15 közötti értek).

2.2.3. Hang megadása és lejátszása

A hang lejátszására szolgál a TUNE utasitás.
Alakja:

TUNE s,l,n,o (,v)

ahol s a megszólaltatandó hang (1., 2. vagy harmadik hang),

l a hang hosszusága 1/60-ad másodpercekben (0-65535 közötti érték),

n a megszólaltatandó hang magassága (értékei 1-12 között):
1:c, 2:cis, 3:d, 4:dis, 5:e
6:f, 7:fis, 8:g, 9:gis, 10:a
11:ais, 12:h

o az oktáv megadására szolgáló paraméter (0-7 közötti érték),

v a hangolásra szolgáló paraméter (értékei 0 és 255 közötti érték lehet).

Példaképpen tekintsük a következő programot:

```
10    VOLUME=15
20    SOUND 1,3,10,15,1,15,1
30    FILTER 1,1000,15
40    FOR J=0 TO 7
50    FOR I=1 TO 12
60    TUNE 1,50,I,J
70    NEXT I
80    NEXT J
```

A fenti program lejátsza a C64 gép minden hangját – összesen 96 hangot – a félhangokat is beleértve.

2.3. 80x50-es grafika

A 80x50-es grafika esetén az x és y értékei a következők lehetnek:

$$0 \le x \le 79$$

$$0 \le y \le 49$$

A képernyő 0,0 pontja a bal felső sarokban van. A fenti számokból látszik, hogy minden egyes betűhely négy részre van felbontva.
A grafika előnye, hogy a normál karaktereket minden további nélkül használhatjuk.

2.4. Rendszer utasitásai

Az alábbiakban összefoglaljuk a rendszer utasitásait. Részletes leirást a rendszerhez adott kézikönyvben találunk.

2.4.1. Rajzoló utasitások

PLOT x,y	x,y koordinátáju pontot megjelenit.
PLOT x1,y1 TO x2,y2	Szakasz rajzolása az x1,y1 pontból az x2,y2 pontba.
PLOT TO x2,y2	Az aktuális grafikus kurzor állásából az x2,y2 pontig egy egyenest huz.
CIRCLE x1,y1,x2,y2	Kör, ellipszis rajzolására szolgáló utasitás, paraméterei: x1,y1 középpont, x2 vizszintes sugár, y2 függőleges sugár.
PAINT v FROM x,y	Alakzatok megszerkesztésére és rajzolására szolgáló utasitás, kezdete az x,y koordinátákkal megadott pont: v = 0 felfelé egy pontnyit v = 1 jobbra egy pontnyit v = 2 lefelé egy pontnyit v = 3 balra egy pontnyit.
PAINT v	A grafikus kurzor aktuális állásától kezdve rajzolja ki a kivánt alakzatot (v értékei mint a fenti utasitásnál).

FRAME d,x1,y1 TO x2,y2	Egy d vastagságu rámát rajzol az x1,y1 bal felső és x2, y2 jobb alsó pontokkal megadott téglalap kerületére.
FILL x1,y1 TO x2,y2	Adott szinnel befestett téglalapot rajzol, amely bal felső csucsának koordinátái x1,y1, jobb alsó csucsának koordinátái x2,y2.
TEXT s,x,y,m	Az x,y koordinátákkal megadott ponttól kezdve kiirja az s szöveget az m móddal: m = 0 nagybetük/grafikus jelek m = 1 nagybetük/kisbetük.

2.4.2. Szinek megadása

FCOL f	Képernyő keretszinének megadása /f = 0 - 15/.
BCOL f	Háttérszín megadása /f = 0 - 15/.
SCOL	Rajzolási szin megadása.
PCOL	Pont szin megadása.

2.4.3. Input-output utasitások

GSAVE n,"f",g	Az f nevü állományba helyezi lemezre /g=8/ vagy szalagra /g=1/ a képernyőn lévő rajzot.

GLOAD n,"f",g	Az f nevü grafikus képernyőt betölti és megjeleniti a g egységről.
HCOPY	Különböző formáival a következő nyomtatókra rajzolhatjuk ki a képernyőtartalmat: - SEICOSHA GP 100VC - EPSON FX-80 /C 1526/ - SEICOSHA GP-700M /8 szinü printer/ - fentiek valamelyikével kompatibilis printer.

2.4.4. Szellem grafika

SREAD s	Az s karaktertipusu változóba elhelyezi a szellem megrajzolásához szükséges adatokat /63 byte gyors olvasása/.
SDEFINE	Szellem megadásához szükséges paramétereknek értékadás.
SMODE s,m,f	Mód megadása, a paraméterek: s a szellem sorszáma m nagyitás és prioritás megadás f szin /szinek/ megadása.
SSET s,x,y	Hely megadására szolgáló utasitás: s a szellem sorszáma x,y a szellem bal felső sarkának koordinátái.

SSET s,x1,y1, TO x2,y2,p	Szellem mozgatására szolgáló utasitás: s sorszámu szellemet az x1,y1 pontból az x2,y2 pontba p sebességgel (1/60 mp egy p értéke, max. érték: p=256).
SSET s TO x,y,p	Mint a fenti utasitás, de a mozgatás a szellem aktuális helyéről történik.
SWAIT s	A program futása az s sorszámu szellem mozgásának végéig áll.
IF par THEN...	Programelágaztató utasitás
RETURN	Megszakitás befejeződik.

2.4.5. Egyéb utasitások

GMODE s ,m	Rajzolás módjának megadására szolgáló utasitás.

s értékei	értelmezése
0	1-es oldal látszik, utasitás erre vonatkozik
1	1-es oldal látszik, utasitás 2-re vonatkozik
2	2-es oldal látszik, utasitás 2-re vonatkozik
3	2-es oldal látszik, utasitás 1-re vonatkozik

m értékei	értelmezése
0	LGR látszik, utasitás erre vonatkozik
1	HRG látszik, utasitás erre vonatkozik
2	MC látszik, utasitás erre vonatkozik
3	HRG látszik, utasitás LGR-re vonatkozik
4	MC látszik, utasitás LGR-re vonatkozik
5	LGR látszik, utasitás HGR-re vonatkozik
6	LGR látszik, utasitás MC-re vonatkozik

ahol LGR 80x50-es grafika
HRG 320x200-as grafika
MC 160x200-as grafika .

GCLEAR	Grafikus képernyő kitörlése.
GMOVE	Grafikus ábrák jobbra /balra/ mozgatására szolgáló utasitás.
ROT=r	A PAINT utasitással kirajzolandó alakzatok forgatására szolgáló utasi - tás /r 0 és 63 közötti érték, azaz 1 r értéke 5.625 fok/
SIZE = s	A PAINT utasitással megadott alakzatok nagyitását végzi az utasitás, /s értéke 0-255 között változhat, s=0 ese - tén nincs nagyitás/.
INVERS	A rajz inverzét adja, azaz azon pontok, amelyek ki voltak "rajzolva", nem lesznek a továbbiakban és viszont!

C64 AIDS

1. Bevezetés

A C64 AIDS a Print-Technik /WIEN/ terméke. Az egyéb segitőkártyához hasonlóan a programozást, programbelövést segiti. Összesen 16 utasitásból áll, ezek közül az egyik a HELP, amely a rendelkezésre álló parancsokat listázza ki. A kártya hasonló segitséget ad, mint a HELP PLUS, a VIC 1212 és a SIMON'S BASIC parancsai. Összehasonlitva a fentiekkel a következőket állapithatjuk meg:

- a HELP PLUS és a SIMON'S BASIC megfelelő utasitásai kényelmesebbek, többet nyujtanak,

- a VIC 1212 egyrészt több, másrészt kevesebb, mint a C64 AIDS, de nem sokkal tér el tőle.

A továbbiakban összefoglaljuk a programrendszer utasitásait.

2. Programozást segitő parancsok

2.1. Programok összemásolása

Programok összemásolását végzi az APPEND parancs.
Alakja:

APPEND "f"

ahol f a másolandó programnév.
Hatása: a gépben levő program végétől kezdve hozzámásolja a lemezen levő f nevü programot.
Igy a másolásnál biztositani kell, hogy a másolandó program kezdő sorszáma nagyobb legyen a gépben levő program legnagyobb sorszámánál.

2.2. Programsorszámok automatikus kiirása

A programozást segiti az AUTO automatikus sorszámozást végző utasitás.
Alakja:

AUTO s,i

ahol s a sorszámozás kezdete,
i az egyes programsorok közötti növekmény.

2.3. Program egyes részeinek eltörlése

A program egyes részeit törli ki a DELETE parancs.

Alakja:

DELETE n,m

ahol n az eltörlendő első programsor sorszáma,
m az eltörlendő utolsó sor sorszáma.

Hatása: Az n. utasitássorszámtól az m.-ig eltörli az utasitásokat - n-t és m-et is beleértve.

2.4. Program ujrasorszámozása

Egy program ujrasorszámozását végzi a RENUMBER parancs.

Alakja:

RENUMBER s,i,n,m

ahol s az átsorszámozandó programrész első utasitásának uj sorszáma,
i az átsorszámozandó programrész utasitás sorszámai közötti különbség,
n az átsorszámozandó programrész első utasitásának sorszáma,
m az átsorszámozandó programrész utolsó utasitásának sorszáma.

2.5. Karakterek sokszorozása

Karakterek sokszorozását végzi el a REPEAT parancs.

Alakja:

REPEAT

Hatása: A parancs kiadása után egy tetszésszerinti billentyüt lenyomva tartva addig irja ki a gép, ameddig fel nem engedjük.

3. Programbelövés segitése

3.1. Változók értékének kiirása

Egy program változói értékeinek kiirását végzi el a DUMP parancs.
Alakja:

DUMP i

ahol i utasitás paraméter a következő értékeket veheti fel:

%, $, (, üres

Hatása:

i = % esetén kilistázza az egész tipusu változók nevét és aktuális értéket,

i = $ esetén kilistázza a karakter tipusu változók nevét és aktuális értékét,

i = (esetén kilistázza a tömbök neveit és megadja a dimenziójukat,

i = üres esetén a fentiek mindegyikét megjeleniti a képernyőn.

3.2. Program stringjeinek kicserélése

Egy program stringjeinek kicserélését végzi az EDIT parancs.
Alakja:

EDIT /s1,s2/ (n-m, Q)

ahol	s1,s2	karakterláncok,
	s1	amit ki akarunk cserélni,
	s2	amire cseréljük s1-et,
	n	opcionális paraméter, a figyelembe veendő programrész első utasitásának sorszáma,
	m	opcionális paraméter, a figyelembe veendő programrész utolsó utasitásának sorszáma,
	Q	ha szerepel, akkor minden s1 megtalálása esetén megkérdezi, hogy cseréljen-e, vagy sem.

Hatása: Az n-edik sorszámu sortól az m-edik sorszámu sorig s1-et s2-re cseréli. Ha Q szerepel, akkor csak "Y" válasz esetén cserél "N" válasz esetén továbbmegy.

3.3. String megkeresése

Egy programban előforduló stringek megkeresését végzi a FIND parancs.

Alakja:

FIND / s / n,m

ahol	s	a megkeresendő string,
	n	az első programsor sorszáma, ahonnan a keresés kezdődik,
	m	az utolsó programsor sorszáma, ameddig a keresés még tart.

Hatása: n-m közötti programsorokban megkeresi az s stringet, és kilistázza azon sorokat, melyekben ez a string előfordul.

3.4. Programfutás nyomonkövetése

A program futásának nyomonkövetését végzi a TRACE utasitás.

Alakja:

TRACE W

ahol W opcionális paraméter.

Hatása:
- Ha a W paraméter nincs kitéve, akkor a program futása lelassul,
- Ha W-t is begépeltünk, akkor az utolsó négy végre - hajtott utasitás sorszámát irja ki a gép a képemyő jobb felső sarkába.

A TRACE utasitást az OFF utasitással lehet hatástalanitani.
Alakja:

OFF

A program lépésenkénti végrehajtását végezhetjük a STEP utasitással.
Alakja:

STEP W

ahol W opcionális paraméter.

Hatása:
- Ha W paramétert nem használjuk, a programot lépésenként hajtja végre és minden utasitás után megáll. Tovább - inditani a SHIFT gombbal lehet.
- Ha a W paraméter is ki van téve, ugy a fentiek mellett az utolsó 4 végrehajtott utasitás sorszámát is kiirja a képernyő jobb felső sarkába.

4. Konstansok konvertálása

Decimális szám konvertálását kettes és tizenhatos számrendszerbeli számba a DEC függvény végzi.
Alakja:

DEC n1 n2 n3 n4 n5

ahol ni decimális számjegyek.

Példa

DEC 64444
=HFBBC
= B 11111011 10111100

Tizenhatos számrendszerbeli számot tizesbe és kettesbe a HEX függvény konvertál át.
Alakja:

HEX h1 h2 h3 h4

ahol hi tizenhatos számrendszerbeli számjegy.

Példa

HEX AAAD
=43693
=B10101010 10101110

5. Egyéb parancsok

5.1. Rendszer kikapcsolása

A rendszer hatását a KILL parancccsal szüntethetjük meg.
Alakja:

KILL

5.2. Eltörölt program visszaállítása

A NEW parancccsal eltörölt programot visszaállitani az OLD parancccsal lehet.
Alakja:

OLD

I. FÜGGELÉK

SIMONS' BASIC	hibaüzenetek
üzenet	jelentése
?BAD MODE	Egy parancs paramétere nem megen - gedett tipusu.
?NOT HEX CHARACTER	Nem hexadecimális számot akartunk decimálissá alakitani.
?NOT BINARY CHARACTER	Nem kettes számrendszerbeli számot akartunk decimálissá alakitani.
?UNTIL WITHOUT REPEAT	Egy UNTIL utasitást használtunk o-lyan helyen, ahol előzőleg nem volt REPEAT utasitás.
?END LOOP WITHOUT LOOP	Egy END LOOP utasitást használ - tunk LOOP utasitás nélkül.
?END PROC WITHOUT EXEC	Egy END PROC utasitást használunk PROC utasitás nélkül.
?PROC NOT EXIST	Egy olyan eljárást kiséreltünk meg be-hivni, amelyik nem létezik.
?NOT ENOUG LINES	Nincs elég sor megadva egy szellem megszerkesztéséhez.
?BAD CHAR FOR A MOB	A szellem megadásánál a DESIGN uta-sitásban a paraméterek nem esnek a meg-adott intervallumba.
?INSERT PARAMETER TOO LARGE	Az INSERT parancs esetén fordul elő ak-kor, ha a hossz megadása nem megfelelő nagyobb, mint a karakterlánc hossza.

?CREATED STRING TOO LONG	Az INSERT utasitással létrejött karakterlánc hossza meghaladja a megengedett 255 karaktert.
BAD MODE	A MOVE utasitással nagyobb területet jelöltünk ki, mint amennyi a képernyőn elfér.

II. F Ü G G E L É K

HELP PLUS parancsai

<u>Programozás, listázás segitése</u>

#G a,b	Automatikus sorszámkiirás a kezdő sorszám b növekmény.
*	Szabad memóriahely kiirása.
#L n	Program listázása képernyőnként n első listázandó sor száma.
#R a,b	Program átsorszámozása a kezdő sorszám b növekmény.
#C a	Program süritése a az egy sorban elhelyezhető karakterek száma.
#A "n", d	Gépben lévő programhoz másik programrész másolása külső tárolóról n másolandó programnév d egységszám, ahol a másolandó program van.

#F s	Egy programrészlet megkeresése s a programban előfordu-ló tetszésszerinti karak-tersorozat.
#D n1-n2	Programsorok törlése n1-től n2-ig eltörli a soro-kat.

Programbelövés segitése

#H	Utolsónak végrehajtott program - sor kiirása megszakitáskor /befe-jezéskor/.
#T	Program végrehajtásának folyama-tos soronkövetése.
#S	Program végrehajtása utasitáson - ként.
#E	#T vagy #S parancs befeje-zése.
#U	Nem megengedett programhivatko-zások /GO TO, GOSUB stb./ el-lenőrzése.
#V	Egyszerü programváltozók tartalmá-nak kiirása képernyőre.

#M	Egy program indexes változóinak értékét kiiró parancs.
!# k1k2k3k4	Tizenhatos számrendszerbeli szám konvertálása tizesbe ki tizenhatos számrendszerbeli számjegy.
! Ø n1n2n3n4n5	Tizes számrendszerbeli szám konvertálása tizenhatosba ni tizes számrendszerbeli számjegy.

Egyéb parancsok

# B	NEW paranccsal eltörölt program visszaállitása.
(d	OPEN és CMD parancs együttes kiadására szolgáló parancs d egységszám.
)	Fenti paranccsal megnyitott file lezárására szolgáló parancs.
£d	Lemez egységszám megváltoztatására szolgáló parancs d uj egységszám.
#K	HELP PLUS befejezésére szolgáló parancs.

Disassembler parancsok

]	Disassembler aktivizálására szolgáló parancs.
+	Cimváltoztatásra szolgáló parancs.
/	Egy byte tartalmának megváltoztatására szolgáló parancs.
[	Nyomtatón való listázás /disassemblálás/ parancsa.
>	Gép memóriájának disassemblálását kijelölő parancs.
<	Floppy egység memóriájának dis - assemblálását kijelölő parancs.
RETURN	Cimet a listázásnál eggyel növeli .
↑	Cimet a listázásnál eggyel csökkenti.
-	A beállitott cimtől folytatódik a disassemblálás.
space	Folyamatos disassemblálás.
CTRL	Kiirás lassitása.
STOP	Kiirás megállitása.

SHIFT	Kiirás átmeneti megállitása.
SHIFT/LOCK	Benyomva a listázás átmenetileg áll, elengedve továbbindul.
*	Gépi kódu program elinditása a disassembláció aktiv állapotában.
@	Memóriaterület átmásolása.
=	Disassemblálás kikapcsolása.

Disk parancsok

/n	Programbetöltés lemezről n betöltendő program neve.
% n	Programbetöltés LOAD "n",8,1 alakban n betöltendő programnév.
↑n	Programbetöltés és futási parancs n a betöltendő és futtatandó program neve.
←n	Programkiirás lemezre n a kiirandó program neve.
<n	Programkiirás ellenőrzése (VERTIFY) n az ellenőrzendő program neve.

$	Lemezdirektori kiirása képernyőre.
@	Disk státusz kiirása.
>	Parancscsatornáról olvasás.

Assembler forditó parancsai

[	Forditó aktivizálása.

III. F Ü G G E L É K

SIMON'S BASIC utasitásai és parancsai

Rajzoló utasitások

HIRES x,y	Átkapcsolás grafikus módra x rajzolás szine y háttérszin.
MULTI x,y,z	Szines grafikus módra való átkapcsolás x 1. rajzolási szin y 2. rajzolási szin z 3. rajzolási szin .
LOW COL p,q,r	További szinek megadása p 1. rajzolási szin q 2. rajzolási szin r 3. rajzolási szin.
HI COL	Visszatérés az eredeti rajzolási szinre.

Geometriai alakzatok megjelenitése

PLOT x,y,p	Egy pont megjelenitése a grafikus képernyőn x,y pont koordinátái p rajzolási tipus.

LINE x1,y1,x2,y2,p	Egy egyenes szakasz kirajzolása	
	x1,y1	szakasz kezdőpontjának koordinátái
	x2,y2	szakasz végpontjának koordinátái
	p	rajzolási tipus.
REC x1,y1,x2,y2,p	Téglalap kirajzolása	
	x1,y1	téglalap bal felső sarkának koordinátái
	x2,y2	téglalap jobb alsó sarkának koordinátái
	p	rajzolási tipus.
CIRCLE x1,y1,x2,y2,p	Kör, ellipszis kirajzolása	
	x1,y1	alakzat középpontjának koordinátái
	x2	alakzat félszélessége
	y2	alakzat félmagassága
	p	rajzolási tipus.
ARC x1,y1,a,b,c,x2,y2,p	CIRCLE utasitással kirajzolható alakzat egy részének kirajzolására szolgáló utasitás	
	x1,y1	u.a.,mint a CIRCLE utasitásnál
	a	rajzolás "kezdetének" szöge
	b	rajzolás végének szöge
	c	lépésköz

	x2,y2	u.a. mint a CIRCLE utasitásnál
	p	rajzolási tipus.
ANGL x1,y1,a,x2,y2,p	CIRCLE utasitással kirajzolható alakzat hurjainak kirajzolása	
	x1,y1	lásd CIRCLE utasitás
	a	függőleges "sugárral" bezárt szög
	x2,y2	u.a. mint a CIRCLE utasitásnál.
DRAW "nnn...9",x,y,p	Grafikus kurzor mozgatása rajzolással vagy anélkül	
	n	mozgatás iránya és rajzolás/nem rajzolás megadása
	x,y	rajzolás kezdetének koordinátái
	p	rajzolási tipus.
RCT s,m	DRAW utasitással megadott alakzat kirajzolása	
	s	forgatás szöge
	m	méret növelési faktor.

Alakzatok szinezése

PAINT x,y,p	Egy zárt alakzat kifestése	
	x,y	az alakzat egy belső pontjának koordinátái
	p	rajzolási tipus.

BLOCK x1,y1,x2,y2,p	Szines téglalap kirajzolása	
	x1,y1	téglalap bal felső sarkának koordinátái
	x2,y2	téglalap jobb alsó sarkának koordinátái
	p	rajzolási tipus.

Rajzok feliratozása

CHAR x,y,c,p,s	Egy karakter kiirása a grafikus képernyőre	
	x,y	karakter kezdetének koordinátái
	c	kirajzolandó karakter POKE kódja
	p	rajzolási tipus
	s	kirajzolandó karakter mérete.
TEXT x,y,"/t/b",p,s,i	Egy szöveg kiirása a grafikus képernyőre	
	x,y	kiirandó szöveg kezdetének koordinátái
	t	alsó/felsőállás vezérlése
	b	kiirandó karaktersorozat
	p	rajzolási tipus
	s	karakter mérete
	i	üres cellák száma a karakterek kezdete között.

További grafikus utasítások

CSET i	Karakterkészletek váltása, utolsó grafikus képernyő megjelenitése i=0 felső állás/grafikus karakterkészlet i=1 alsó/felső állás i=2 legutolsó grafikus képernyő kirajzolása.
TEST (x,y)	Megadott pont állapotának ellenőrzése x,y megadott pont koordinátái.

BASIC utasitások kiterjesztése

IF...THEN...ELSE...	Feltételes utasitás kiterjesztése.
RCOMP...ELSE...	Az utasitást megelőző IF...THEN ...ELSE utasitásra való hivatkozás.
REPEAT...UNTIL L	Ciklusutasitás, a REPEAT UNTIL közötti utasitásokat addig hajtja végre, ameddig L igaz nem lesz, L logikai kifejezés.
LOOP...EXIT IF L END LOOP	A LOOP és EXIT közötti utasitásokat addig hajtja végre, ameddig L értéke igaz nem lesz, L logikai kifejezés.

c1 PROC n... END PROC	Zárt szubrutin, c1 kezdőcimke n szubrutin név END PROC szubrutin végét jelzi.
c PROC n	Cimke megadása c utasitáscimke n szimbolikus név.
CALL n	Előző utasitással megadott helyre való ugrás n szimbolikus név.
c LOCAL l	Helyi változók megadása c utasitáscimke l helyi változók listája.
c GLOBAL l	Előző utasitással helyinek deklarált változókat visszaállitja eredeti értékükre c utasitáscimke l globális változók listája.
CGOTO e	e aritmetikai kifejezés egész értékére történő ugrás.
RESET c	DATA utasitás aktuális pointerének megváltoztatása c egy DATA utasitás cimkéje.

Irás-olvasás utasitásai

CENTRE "s"	Képernyőre kiirandó szöveg középre helyezése s kiirandó karakterlánc.
USE "p.q",v:PRINT	v változó értékének kiirása p egész és q tizedes jegyre p ésq jelek sorozata
USE "s1p.qs2",v:PRINT	v változó értékének kiirása s1 "kezdőszöveggel", p egész q tizedes jegyre s2 befejező szöveggel pésq jelek sorozata s1,s2 tetszésszerinti karakter - lánc.
PRINT AT (p,q)"s"	p. oszlop q. elemétől kiirja az s karakterláncot.
PRINT AT (p,q)v	p. oszlop q. elemétől kiirja a v változó értékét.
FETCH "v",n,vn	vn változóba n karakter beolvasása a v vezérlő-karakternek megfelelően.
INKEY	Numerikus értékü függvény, értéke a lenyomott funkció billentyü száma /1-16/vagy 0.

Egyéb utasitások

PAUSE "u",i	Programfutás felfüggesztése az u karaktersorozat kiirásával i másodpercre.
PAUSE i	Programfutás felfüggesztése i másodpercre.
ON ERROR: GO TO c	1-5 és 10-23 sorszámu hibák programból lekezelése c utasitáscimke ERRLN hibás utasitás sorszáma ERRN hiba kódja.
NO ERROR	Fenti hibák lekezelése után a futás továbbinditása.
DISAPA:u	Program utasitássorának előkészitése titkositásra u titkositandó utasitás.
SECURE 0	Titkositás elvégzésére szolgáló parancs.

BASIC függvények kiterjesztése

MOD (a,b) — Modulus függvény: a moduló b.

DIV (a,b) — Egészrész osztás
- a,b — egész vagy valós számok.

FRAC (k) — k aritmetikai kifejezés tizedes része.

EXOR (a,b) — Kizáró vagy a és b számok /bináris alakja/ esetén.

INSERT ("s1","s2",n) — Karaktersorozat beszurása
- s1 — beszurandó karak - tersorozat
- s2 — karakterlánc, ami- be a beszurás tör - ténik
- n — azon pozició helye, ahonnan a beszurás történik s2-ben.

INST ("s1","s2",n) — Karakterlánc végének megváltoz- tatása
- s2 — karakterlánc, a végét változtatjuk meg
- s1 — az s2 végére irandó karakterlánc
- n — azon karakter pozici- ójának helye, ahon - nan a változtatás tör- ténik s2-ben.

PLACE ("s1","s2")	Karakterlánc helyének megkeresése s1 keresendő karakterlánc s2 karakterlánc, amiben keresünk.
DUP ("s",n)	Karakterlánc többszörözése s többszörözendő karakterlánc n többszörözés paramétere.
%b	Bináris szám konvertálása tizedesbe b bináris szám.
Øh	Tizenhatos számrendszerbeli szám konvertálása tizesbe h hexadecimális szám.
PENX	Fényceruza aktuális helyének x koordinátája.
PENY	Fényceruza aktuális helyének y koordinátája.
POT (i)	Paddle potenciométerének aktuális állása i berendezés száma /0 v. 1/.
JOY	Botkormány és a tüzelőgomb állásának meghatározása.

Képernyő műveletek

FLASH sz,s	Képernyő szineinek villogtatása sz — szinkód s — sebesség.
OFF	Képernyő villogtatásának befejezése.
BFLASH s,sz1,sz2	Képernyő keretének megadott szinekkel történő villogtatása s — sebesség sz1,sz2 — villogtatandó szinek kódjai.
BFLASH 0	Keret szinvillogtatásának befejezése.
FCHR a,b,c,d,e	Képernyő adott területének adott karakterekkel való feltöltése a — kezdő sor sorszáma b — kezdő oszlop oszlopszáma c — terület szélessége karakterekben d — terület "mélysége" karakterekben e — megjelenitendő karakter POKE kódja.

FCOL a,b,c,d,e

Képernyő adott területén levő jelek szinének megváltoztatása

- a-d megegyezik az FCHR utasitásnál elmondottakkal
- e terület karaktereinek szine.

FILL a,b,c,d,e,f

Adott képernyő-terület adott szinü karakterekkel való feltöltése

- a-e megegyezik az FCHR utasitásnál elmondottal
- f területen megjelenitett karakter szin kódja.

MOVE a,b,c,d,e,f

Képernyő aktuális tartalmának átmásolása a képernyő más területére

- a-d megegyezik az FCHR utasitásnál elmondott paraméterekkel
- e az uj terület bal felső sarkának sorindexe
- f az uj terület bal felső sarkának oszlopindexe.

INV a,b,c,d

Karakterek inverz képének megjelenitési helyét jelöli ki

- a-d területet megadó paraméterek /lásd FCHR utasitás/.

LEFT i,a,b,c,d

Képernyő egy területének rollozása balról jobbra

- i — tekerés tipusa
- a,b — mozgatási terület bal felső sarkának oszlop ill. sor indexe
- c,d — mozgatási terület jobb alsó sarkának oszlop ill. sor indexe

RIGHT i,a,b,c,d

Képernyő egy területének rollozása jobbról balra

- i-d — paramétereket a LEFT utasitásnál ismertettük.

UP i,a,b,c,d

Képernyő egy területének rollozása alulról felfelé

- i-d — paramétereket a LEFT utasitásnál ismertettük.

DOWN i,a,b,c,d

Képernyő egy területének rollozása felülről lefelé

- i-d — paramétereket a LEFT utasitásnál ismertettük.

SCRSV 2,8,2,"n,S,W"

Képernyő aktuális tartalmának elhelyezése mágneslemezen

- n — adatállomány neve.

SCRSV 1,1,1,"n,S,W"	Képernyő aktuális tartalmának elhelyezése mágnesszalagon n adatállomány neve.
SCRLD 2,8,2,"n"	Képernyő adatainak visszaolvasása lemezről, majd megjelenitése n adatállomány neve.
COPY	Karakteres vagy grafikus képernyő tartalom kiirása nyomtatón.
HRDCOPY	Karakteres képernyő tartalom kiirása nyomtatón.
LIN	Kurzor aktuális sorszámának meghatározása.

Programirást segitő parancsok

AUTO s,n	Automatikus program-sorszámok generálása s kezdő sorszám n növekmény.
RENUMBER s,n	Gépben lévő program átsorszámozása s uj kezdősorszám n növekmény.

PAGE n	Program listázása képernyőre részenként n egyszerre a képernyőre kiirandó sorok száma.
DELAY n	Program listázásának lassitása n listázás sebessége.
OPTION n	SIMON'S BASIC utasitások kiírása forditott /REVERSE/ módban /n=10 esetén/ n ≠ 10 és 0 és 255 közötti érték: normál listázás.
FIND c	c karakterlánc előfordulási helyének /helyeinek/ megkeresése.

Szellem és karakter készitése

DESIGN a,b	Helyfoglalás egy szellem /sprite/ részére a tipus b tároló első byte-jának cime.
@	Egyszinű, vagy szines szellem illetve saját karakter megszerkesztéséhez használatos séma.
CMOB s1,s2	Szinek megadása multikolor módban s1,s2 szinkódok.

MOB SET a,b,c,d,e	Szellem paramétereinek megadása a szellem száma b blokkszám c szinkód d prioritás e tipus kijelölése.
MMOB a,x1,y1,x2,y2,b,c	Szellem megjelenitése és mozgatása a szellem száma x1,x2 megjelenités kezdetének koordinátái x2,y2 mozgatás végpontjának koordinátái b szellem mérete c mozgatási sebesség.
RLOCMOB a,x,y,b,c	Képernyőn levő szellem mozgatása a szellem száma x,y mozgatás végpontjának koordinátái b,c megegyeznek a MMOB utasitásnál elmondottakkal.
DETECT n	Alakzatok összeütközésének lekérdezését késziti elő n alakzat meghatározása.

IF CHECK (m1,m2)=0 THEN u	Összeütközések lekérdezése m1,m2 szellemek száma u utasitás.
IF CHECK(0)=0 THEN u	Szellem és karakterek összeütközésének lekérdezése.
MOB OFF	Megjelenitett szellem kitörlése a képernyőről.
MEM	Karakterkészlet átmásolása ROM--ból RAM-ba.
DESIGN 2,$E000+c*8	Kicserélendő karakter megadása c karakter PÓKE kódja.

Zenei utasitások

VOL n	Hangerő megadása.
WAVE h,b	Hang megadása h hang száma b hangjellemzők paraméterei.
ENVELOPE h,A,D,S,T	Burkoló görbe megadása h hang száma

	A felfutás maximuma
	D lefutás egyenletes erősségre
	S egyenletes erősség ideje
	T lefutás nullára.
MUSIC n,"c"	A zene tempójának és hang-jainak megadása n tempó paraméter c zenét megadó karakterlánc.
PLAY n	Zene lejátszása n lejátszás módja.

<u>Disk parancsok</u>

DISK,"müvelet"	Disk müveletek megadása.
DIR "Ø....	Disk állományok neveinek kiírása.

<u>További parancsok</u>

OLD	NEW parancs hatástalanitása.

KEY s,"c"	Funkció billentyűk aktivizálása s billentyű száma c utasitásnév.
DISPLAY	Funkció billentyű értékének ki - irása.

IV. F Ü G G E L É K

SUPERGRAPHIK 64 szalagos változat utasitásai

!GCLEAR	Grafikus képernyő kitörlése.
!GRAPHICS p	Grafikára való átváltás és tipus-kijelölés p rajzolási paraméter.
!DISP p,x,y,s	Egyenesekből összeállitott alakzat kirajzolása p rajzolási paraméter x,y rajzolás kezdetének koordinátái s kirajzolandó alakzatot megadó karakterlánc.
!DOT x,y	Egy megadott koordinátáju pont kirajzolása x,y megjelenitendő pont koordinátái.
!CDOT x,y	Egy megadott koordinátáju pont törlése x,y kitörlendő pont koordinátái
!TEST x,y	Egy megadott koordinátáju pont állapotának tesztelése

!LINE x1,y1, TO x2,y2

Egyenes szakasz rajzolása két koordináta pontokkal megadott pont között

- x1,y1 kezdőpont koordinátái
- x2,y2 végpont koordinátái.

!CLINE x1,y1 TO x2,y2

Egyenes szakasz törlése

- x1,y1 kezdőpont koordinátái
- x2,y2 végpont koordinátái.

!DLINE x1,y1, TO x2,y2

Egyenes szakasz kirajzolása pontozva a megadott pontok között

- x1,y1 kezdőpont koordinátái
- x2,y2 végpont koordinátái.

!FILL x1,y1 TO x2,y2

Téglalap kirajzolása és befestése

- x1,y1 bal felső sarok koordinátái
- x2,y2 jobb alsó sarok koordinátái.

!CLEAR x1,y1 TO x2,y2

FILL utasítással rajzolt alakzat törlése

- x1,y1 bal felső sarok koordinátái
- x2,y2 jobb alsó sarok koordinátái

!FRAME x1,y1 TO x2,y2

Két koordináta ponttal megadott téglalap oldalainak kirajzolása

- x1,y1 bal felső sarok koordinátái
- x2,y2 jobb alsó sarok koordinátái.

!CFRAME x1,y1 TO x2,y2	FRAME utasítással kirajzolt téglalap törlése x1,y1 bal felső sarok koordinátái x2,y2 jobb alsó sarok koordinátái.
!MOVE x,y	Grafikus kurzor mozgatása rajzolás nélkül a megadott pozicióba x,y koordinátákkal megadott pontba.
!DRAW x,y	Egyenes szakasz rajzolása a kurzor aktuális poziciójától kezdve a megadott pontig x,y végpont koordinátái.
!GSAV "n",d	Aktuális grafikus képernyőtartalom elhelyezése mágneslemezre vagy mágnesszalagra n állomány neve d egységszám.
!RECALL "n",d	GSAV utasítással háttértárolóra kiírt grafikus képernyőtartalom beolvasása és megjelenitése n állomány neve d egységszám.

!HARD n	Aktuális grafikus képernyőtartalom kiirása sornyomtatón n nyomtató száma.
!TRANS	A két grafikus képernyőtartalom közötti felcserélő utasitás.
!SPRITE n,v	Szellem adatainak megadása n szellem száma v szellem adatait tartalmazó változó.
!SMODE n,s,p	Szellem méretének és szinének megadása n szellem száma s nagyság p szellem szine.
!SMOVE n,x,y	Szellem mozgatása n szellem száma x,y végpont koordinátái.
!BCOL n	Képernyő háttérszinét megadó utasitás n háttérszin kódja.
!LCOL n	Grafikus rajzolási szin megadása n rajzolási szin kódja.

!TEXT "s",x,y,p,/,c/

Szöveg kiirása /és eltörlése/ grafikus képernyőre /képernyőről/

s	kiirandó szöveg
x,y	kiirás kezdetének koor - dinátái
p	karakterkészlet
c	szöveg szine.

!CIRCLE x1,y1,x2,y2,/s1,s2/

Megadott középpontu, szélességü és magasságu kör /ellipszis/ kirajzolása

x1,y1	alakzat középpontjának koordinátái
x2	x irányu sugár hossza
y2	y irányu sugár hossza
s1	a rajzolás kezdőszöge fokokban
s2	a rajzolás "vég"-szöge fo - kokban.

!DCIRCLE x1,y1,x2,y2,/s1,s2/

Megadott középpontu, szélességü és magasságu kör /ellipszis/ kirajzolása pontozással

a paraméterek megegyeznek a CIRCLE utasitásnál elmondottakkal.

!CCIRCLE x1,y1,x2,y2,/s1,s2/

Grafikus képernyőre kirajzolt kör /ellipszis/ eltörlése

a paraméterek megegyeznek a CIRCLE utasitásnál elmondottakkal.

!OFF

Rendszer lekapcsolása.

V. F Ü G G E L É K

SUPERGRAPHIK 64 lemezes változat utasitásai

Zenei utasitások

VOLUME = v	Hangerő beállitása.
SOUND s,w,a,d,p,r,(f(p))	Hangjellemzők megadása s hang sorszáma w hullámforma a felfutás nulláról maximumra d lefutás átlagszintre p átlagszint ideje r lefutás nullára f(p) szűrő.
FILTER a(,b,c)	Szűrő paramétereinek beállitása a szűrő jellemző megadása b szűrő határfrekvencia c szűrő rezonancia frekvencia.
TUNE s,l,n,o(,v)	Hang lejátszása s hang sorszáma

l	hang hossza
n	magasság
o	oktáv szám
v	hangolási paraméter.

Rajzoló utasitások

PLOT x,y — Az x,y pont megjelenitése.

PLOT x1,y1 TO x2,y2 — Szakasz kirajzolása

x1,y1	kezdőpont koordinátái
x2,y2	végpont koordinátái.

PLOT TO x,y — Egyenes rajzolása az x,y pontig.

CIRCLE x1,y1,x2,y2 — Kör /ellipszis/ rajzolása

x1,y1	középpont koordinátái
x2	x irányu sugár hossza
y2	y irányu sugár hossza.

PAINT v FROM x,y — Egyenes szakaszokból összeálli-tott alakzatok kirajzolása

v	irány
x,y	rajzolás kezdetének koordinátái.

PAINT v — Fenti alakzat kirajzolása a kurzor aktuális helyétől kezdve.

FRAME d,x1,y1 TO x2,y2	Egy téglalap alaku ráma rajzolása d ráma vastagsága x1,y1 bal felső sarokpont koordinátái x2,y2 jobb alsó sarokpont koordinátái.
FILL x1,y1 TO x2,y2	Szines téglalap kirajzolása x1,y1,x2,y2 megegyezik a FRAME utasitásnál elmondottakkal.
TEXT s,x,y,m	Szöveg kiirása s szöveg x,y kiirás kezdetének koordinátái m karakterkészlet.

Szinek megadása

FCOL f	Keretszin megadása f szinkód.
BCOL f	Háttérszin megadása f szinkód.
SCOL	Rajzolási szin megadása.
PCOL	Pont szin megadása.

Szellemgrafika

SREAD s	Szellem adatainak beolvasása	
	s	karakter tipusu változó.
SDEFINE	Szellem paramétereinek értékadás.	
SMODE s,m,f	Mód megadása	
	s	szellem sorszáma
	m	nagyitás és prioritás
	f	szellem szine.
SSET s,x,y	Megjelenités helyének megadása	
	s	szellem sorszáma
	x,y	megjelenités helyének koordinátái.
SSET s,x1,y1 TO x2,y2,	Szellem mozgatása	
	s	szellem sorszáma
	x1,y1	kiindulási pont koordinátái
	x2,y2	végpont koordinátái
	p	mozgatási sebesség.

SSET s TO x,y,p	Szellem mozgatása az x,y pontba s szellem sorszáma p mozgatási sebesség.
SWAIT s	Program futásának felfüggesztése az s szellem mozgatásának idejére.
RETURN	Megszakitás befejezése.

Egyéb utasitások

GSAVE n,"f",g	Képernyőtartalom elhelyezése külső tárolón f állomány neve g tároló egységszáma.
GLOAD n,"f",g	Rajz betöltése és megjelenitése paraméterek megegyeznek a GSAVE-nél elmondottakkal.
GMODE s,m	Rajzolási mód megadása s utasitás képhez rendelése m rajzolás finomsága.
GCLEAR	Grafikus képernyő kitörlése.

GMOVE	Grafikus ábrák mozgatása.
ROT = r	PAINT utasitással kirajzolt alakzatok forgatása r forgatás szöge.
SIZE = r	PAINT utasitással kirajzolt alakzatok nagyitása r nagyitás mértéke.
INVERS	Grafikus kép inverzének megjelenitése.

IV. F Ü G G E L É K

C 64 AIDS parancsai

Programozást segitő parancsok

APPEND "f"

Programok összemásolása

- f másolandó program neve.

AUTO s,i

Programsorszámok automatikus generálása

- s sorszámozás kezdete
- i növekmény.

DELETE n,m

Programrész törlése

- n törlendő első utasitás sorszáma
- m törlendő utolsó utasitás sorszáma.

RENUMBER s,i,n,m

Programrész átsorszámozása

- s átsorszámozandó programrész első utasitásának uj sorszáma
- i növekmény
- n programrész kezdő utasitásának sorszáma
- m programrész utolsó utasitásának sorszáma.

REPEAT

Billentyűhöz tartozó karakterek több-

Programbelövés segitése

DUMP i	Programváltozók értékének kiirása i tipust megadó paraméter.
EDIT /s1,s2/ (n-m,Q)	Program egy karakterláncának ki - cserélése s1 amit cserélünk s2 amire cserélünk n első utasitás sorszáma m utolsó utasitás sorszáma Q csere vezérlése.
FIND /s/ n,m	Karakterlánc megkeresése s keresendő karakterlánc n,m az EDIT-nél megadott pa - raméterek.
TRACE W	Program végrehajtásának nyomon követése W opcionális paraméter.
OFF	TRACE utasitás kikapcsolása.
STEP W	Program utasitásonkénti végigkövetése W opcionális paraméter.

Konstansok konvertálása

DEC n1... Decimális szám konvertálása tizenhatosba és kettesbe
- ni tizes számrendszerbeli számjegy.

HEX h1... Tizenhatos számrendszerbeli szám konvertálása tizesbe és kettesbe
- hi tizenhatos számrendszerbeli számjegy.

Egyéb parancsok

KILL Rendszer kikapcsolása.

OLD NEW paranccsal eltörölt programok visszaállitása.

T a r t a l o m j e g y z é k

oldal

oldal

oldal

oldal

oldal

commodore 64

105 Ft.